

DOKTORI ÉRTEKEZÉS

Hoffmanné Szalay Zsófia

**Dinamikus NMR spektrumok szimulációjának
elmélete és gyakorlata csatolt spinrendszerekben**

Témavezető: Rohonczy János
egyetemi docens

ELTE Kémiai Doktori Iskola
Doktori iskola vezetője: Inzelt György

Elméleti és fizikai kémia, anyagszerkezetkutatás doktori program

Programvezető: Surján Péter



Eötvös Loránd Tudományegyetem

Természettudományi Kar

Budapest, 2010.

Köszönetnyilvánítás

Szeretnék köszönetet mondani témavezetőmnek, Rohonczy Jánosnak, a lelkes támogatásért és a rengeteg segítségért, amit PhD munkám során adott. Külön köszönöm, hogy felhívta a dolgozatban tárgyalt témára a figyelmemet.

Emellett szeretném köszönetemet kifejezni mindenkinek, akik PhD munkámat jó tanáccsal, kérdéseikkel vagy érdeklődésükkel segítették.

Tartalomjegyzék

Jelölések	7
1. fejezet Bevezetés	11
2. fejezet A szimulációs módszerek irodalmi háttére	15
2.1. Monte Carlo módszer	16
2.1.1. A legismertebb Monte Carlo szimulációk	16
2.1.2. A kinetikus Monte Carlo módszer	17
2.2. A GPGPU alapú programozás	18
2.2.1. A CUDA alapjai	18
2.2.2. Néhány példa a GPGPU alkalmazására	21
3. fejezet Dinamikus NMR spektroszkópia	23
3.1. A dinamikus NMR jelenség	24
3.2. A dinamikus NMR jelenség értelmezése	25
3.2.1. Az egymagos vektormodell	25
3.2.2. A dinamikus NMR jelenség értelmezése csatolatlan spinekre	26
3.3. NMR spektrumok szimulációjának gyakorlata	28
3.3.1. Statikus NMR spektrumok szimulációja	28
3.3.2. Dinamikus NMR spektrumok szimulációja	31
4. fejezet Elméleti eredmények	35
4.1. Alapfogalmak	36
4.1.1. A spinhalmaz	36
4.1.2. Az egyedi sűrűségmátrix	38
4.2. A cserefolyamatok kinetikai leírása	39
4.3. A <i>fid</i> szimulációja	41
4.3.1. Precesszió (propagálás)	41
4.3.2. Cserefolyamatok	42
4.3.3. Detektálás	43
4.3.4. A spektrum direkt szimulációja	44
4.4. Lehetséges egyszerűsítések	46
4.4.1. Szimuláció a Hilbert-térben	46
4.4.2. Transzformált sűrűségmátrix	48

4.4.3.	Blokkdiagonalizáció	48
4.4.4.	A Hilbert- és Liouville térbeli leírás ekvivalenciája.....	49
4.5.	Az átlag- és az egyedi sűrűségmátrix szimulációjának összevetése.....	52
4.5.1.	Mátrixok mérete	52
4.5.2.	Egy kétfasos kölcsönös csere folyamat.....	53
4.6.	A vektormodell és a dinamikus folyamatok	57
4.6.1.	Csatolt spinrendszerek a vektormodellben.....	57
4.6.2.	A populációkról.....	61
5. fejezet	Gyakorlati megvalósítás: ProMoCS	65
5.1.	A ProMoCS program szerkezete	66
5.1.1.	A számolás algoritmus.....	66
5.1.2.	Megosztás több processzoron.....	68
5.2.	ProMoCS-CUDA.....	69
5.2.1.	Nagy sűrűségmátrixok.....	70
5.2.2.	Kis sűrűségmátrixok.....	72
5.3.	Teszteredmények	75
5.3.1.	Tesztelési módszer.....	75
5.3.2.	Futási időt meghatározó paraméterek.....	76
5.3.3.	CPU és GPU összehasonlítása.....	78
6. fejezet	Alkalmazási példák.....	81
6.1.	Egy nagy spinrendszer.....	82
6.2.	Egy erősen csatolt spinrendszer.....	84
6.3.	Csatolt spinrendszer kölcsönös cserével.....	85
6.4.	Xantfoszfát ligandum.....	86
Összefoglalás		89
Abstract		91
Függelék.....		93
F.1	A teszteléshez használt spinrendszerek adatai.....	94
F.2	A teszt futásidők	95
F.3	A teszt spektrumok	97
Irodalomjegyzék.....		99
A dolgozat alapjául szolgáló publikációk		100
Irodalmi hivatkozások		100

Jelölések

J.1 Általános

i	komplex egység.
h_p	Planck-állandó.
k_B	Boltzmann-állandó.
R	egyetemes gázállandó.
m	konformerek száma.
n	spinhalmoz mérete.
n'	spinhalmoz szorzatfüggvényeinek száma, $n' = 2^n$.
N	a rendszerben lévő magok száma, $N \approx n \cdot m$.
M	az n magból álló spinrendszer egykvantum-koherenciáinak száma.
t	a detektálás kezdete óta eltelt (globális) idő.
ω	precesszási körfrekvencia.
μ, μ'	a magok indexe.
ν_μ	a μ -edik mag kémiai eltolódásának megfelelő frekvencia (Hz-ben).
$J_{\mu\mu'}$	a μ -edik és μ' -edik magok közötti csatolási állandó (Hz-ben).
H	Hamilton-operátor mátrixa.
$fid^{(r)}(t)$ ill. $FID(t)$	egy időszelvény, illetve egy spinhalmoz fid -je a t időpontban, mint komplex szám.
$F(t)$	a szimulált fid a t időpontban.
$Y(\omega)$	a szimulált spektrum értéke ω frekvenciánál.
k^*	a jelek látszólagos relaxáció sebessége, a látszólagos relaxációs állandó (T_2^*) reciproka.
ε	egyenletes eloszlással generált véletlen szám.

J.2 Indexek

a, b, c, d	a Hilbert-tér bázisfüggvényeinek indexei. Lehetséges értékei $1 \dots 2^n$.
j, k	a Hilbert-térben a Hamilton-operátor sajátfüggvényeinek indexei. Lehetséges értékei $1 \dots 2^n$.

e, f	a Liouville-tér bázisfüggvényeinek indexei. Lehetséges értékei $1 \dots M$. Hiperindexként $e = (a, b)$ és $f = (c, d)$.
p	a Liouville-térbeli sajátfüggvények indexe. Lehetséges értékei $1 \dots M$. Hiperindexként $p = (k, j)$.
g, h	a konformerek indexe. Lehetséges értékei $1 \dots m$.
l	a totálspinkvantszám szerinti alterek és blokkok indexe. Lehetséges értékei $0 \dots m$ a Hilbert- és $1 \dots m$ a Liouville-térben.
r	az időszeletek és csereidőpontok indexe. Az r -edik időszelet az $r - 1$ -edik cserével kezdődik és az r -edik cserével ér véget.
s	scanek indexe.

J.3 Kinetika

$A^{(h)}$ ill. $[A^{(h)}]$	a molekula lehetséges konformerei, illetve azok koncentrációja.
k_{gh}	a $A^{(g)} \rightarrow A^{(h)}$ cserefolyamat sebességi együtthatója.
K_h	az $A^{(h)}$ konformer relatív egyensúlyi koncentrációja.
d_h	az $A^{(h)}$ konformer bomlási együtthatója.
κ_{gh}	az $A^{(g)}$ konformer bomlása esetén a $A^{(h)}$ konformer keletkezésének valószínűsége.
$t^{(r)}$	az r -edik csere időpontja.
Δt	az utolsó csere óta eltelt idő, azaz $\Delta t = t - t^{(r-1)}$.
$\Delta t^{(r)}$	az r -edik időszelet hossza, azaz $\Delta t^{(r)} = t^{(r)} - t^{(r-1)}$.
$h^{(r)}$ ill. h_r	az r -edik időszeletben jelen lévő konformer indexe.
T	hőmérséklet.
ΔH_{gh}^{++} ill. ΔS_{gh}^{++}	az $A^{(g)} \rightarrow A^{(h)}$ cserefolyamat aktiválási entalpiája és entrópiája.

J.4 Hilbert-tér

$H^{(r)}$	a Hamilton-operátor a Hilbert-térben az r -edik időszeletben.
φ_a	a Hilbert-tér szorzatfüggvényei ($\alpha\alpha\dots\alpha, \beta\alpha\dots\alpha, \dots, \beta\beta\dots\beta$).
$\psi_k^{(h)}, \psi_k^{(r)}$	az $A^{(h)}$ konformer, illetve r -edik időszelet Hamilton-operátorának k -adik sajátfüggvénye a Hilbert-térben.

$\rho(t)$ ill. $\rho_{ab}(t)$	a sűrűségmátrix, illetve ennek (a,b) eleme a Hilbert-térben a φ függvényeken kifejtve.
$\rho^{(r)}$ ill. $\rho_{ab}^{(r)}$	a sűrűségmátrix, illetve annak elemei az r -edik csere időpontjában, azaz $\rho(t^{(r)})$ ill. $\rho_{ab}(t^{(r)})$ röviden.
$\xi^{(r)}(t)$ ill. $\xi_{ij}^{(r)}(t)$	a sűrűségmátrix, illetve annak elemei az r -edik időszaklet konformerének sajátfüggvényeinek bázisán.
$\lambda_k^{(r)}, A^{(r)}$ ill. $A_l^{(r)}$	a Hilbert-térbeli Hamilton-operátor k -adik sajátértéke, az ezekből felépített diagonális mátrix, illetve ennek az l indexű altérhez tartozó blokkja.
$u_{ak}^{(r)}, U^{(r)}$ ill. $U_l^{(r)}$	a $\psi_k^{(r)}$ függvény φ_a -hoz tartozó lineárkombinációs együtthatója, azaz $\langle \psi_k^{(r)} \varphi_a \rangle$, az ezekből felépített unitér mátrix, illetve ennek l -edik altérhez tartozó blokkja.
$a_{kj}^{(r)}, A^{(r)}$ ill. $A_l^{(r)}$	a r -edik időszakletben jelen lévő konformer $\psi_k^{(r)} \rightarrow \psi_j^{(r)}$ átmenet-ének amplitúdója, az ezekből felépített mátrix, illetve ennek l -edik altérhez tartozó blokkja.
Y_{kj}	$\psi_k \rightarrow \psi_j$ átmenet intenzitása, azaz a_{kj}^2 .
I_{ab}	a kiválasztási szabályt kódoló mátrix elemei. A mátrixelem értéke 1, ha a $\varphi_a \rightarrow \varphi_b$ megengedett, egyébként 0.
J.5 Liouville-tér	
$L^{(r)}$	a Hamilton-operátornak megfeleltethető mátrix a Liouville-térben.
$P^{(r)}(\Delta t)$	a sűrűségmátrixot Δt idővel propagáló operátor.
Φ_e	a $\varphi_a \rightarrow \varphi_b$ szorzatfüggvények által meghatározott koherencia.
$\Psi_p^{(h)}, \Psi_p^{(r)}$	az $A^{(h)}$ konformer ill. r -edik időszaklethez tartozó operátor saját-függvényei a Liouville-térben.
$\sigma(t)$ ill. $\sigma_e(t)$	a sűrűségmátrix vektora a Liouville-térben a t időpontban illetve annak e -edik eleme.
$\sigma^{(r)}$ ill. $\sigma_e^{(r)}$	a sűrűségmátrix vektora, illetve annak elemei az r -edik csere idő-pontjában, azaz $\sigma(t^{(r)})$, illetve $\sigma_e(t^{(r)})$ röviden.

$\zeta^{(r)}(t)$ ill. $\zeta_p^{(r)}(t)$	a sűrűségmátrix vektora, illetve annak p -edik eleme az r -edik időszelét Hamilton-operátorának sajátfüggvényeinek bázisán.
$\omega_p^{(r)}$ ill. $\Omega^{(r)}$	az r -edik időszelét konformerének p -edik karakterisztikus frekvenciája, illetve az ezekből felépített diagonális mátrix.
$q_p^{(r)}$ ill. $Q^{(r)}$	az r -edik időszelét konformerének p -edik sajátfüggvénye által adott jel amplitúdója, illetve az ezekből felépített sormátrix.
$c_{ep}^{(r)}$ ill. $C^{(r)}$	az r -edik időszelét konformerének lineárkombinációs együtthatói a Liouville-térben, illetve az ezekből felépített mátrix.
I^+ ill. ϕ_e	a gerjesztő operátornak megfelelő sorvektor a Φ_e koherenciák bázisán, illetve ennek e -edik eleme.

J.6 CUDA

d_x, d_y	a CUDA szálblokkok dimenziója.
b	egy blokk indexe a <i>grid</i> -en belül.
p_x, p_y	egy szál indexe a CUDA blokkon belül.
g	a CUDA <i>grid</i> mérete (blokkokban).
s_i ill. n_f	a <i>grid</i> -en, illetve egy blokk által egyszerre számolt scanek száma
i_x, i_y	a szál által számolt mátrixelem indexei a mátrixban.
i_m	a szál által számolt mátrix indexe.
Y , ill. Y_R, Y_I	a megosztott memóriában lefoglalt mátrix, illetve annak valós és képzetes része.
X	a kinetikai paramétereket tartalmazó mátrix.
ξ, ξ_{temp}	az átmeneti eredmények tárolására használt mátrix a globális memóriában.
ft	a végeredmény tárolására használt mátrix a globális memóriában.

1. fejezet

Bevezetés

A gyakorlati NMR spektroszkópiában gyakran találkozunk dinamikus magcsere-folyamatokkal [5–11]. Ilyen például savas protonok cseréje vizes (vagy savas) oldatban, vagy a klasszikus vonalkiszéledési és koaleszcencia jelenség. A vonalkiszéledés szimulációja lehetőséget ad a kérdéses cserefolyamat kvantitatív vizsgálatára is. Ezt a módszert az NMR spektroszkópia és a kémia számos területén alkalmazzák, így a komplexek stabilitásának és izomerizációjának vizsgálatára, vagy fehérjék dinamikájának tanulmányozására. Az egyes témákról és magáról a dinamikus NMR jelenségről is számos áttekinthető könyv és cikk jelent meg az NMR spektroszkópia kezdetétől egészen napjainkig [12–18].

A dinamikus NMR spektrumok szimulációja (kémiai eltolódások és csatolási állandók alapján) 10-15 magnál nagyobb rendszerekre már csak speciális esetekben (pl. molekulaszimmetriát figyelembe vevő egyszerűsítésekkel [19]) lehetséges. A probléma alapvetően a szimuláció memóriaigénye, ezért a spektrumszimuláló programok fejlesztésének fő iránya a sűrűségmátrix effektív méretének csökkentésére irányul (pl. triviálisan nulla elemek elhagyása, szimmetria kihasználása) [20, 21].

PhD munkám során a dinamikus NMR spektrumok egy kinetikus Monte Carlo szimuláción alapuló számolásának elméletét dolgoztam ki, amelyben a spinrendszer megközelítése alapvetően eltér a hagyományos módszerekétől. Ez a megközelítés a cserefolyamatok félig fenomenologikus - félig kvantummechanikai értelmezése helyett a rendszer statisztikus viselkedését modellezi. Az elmélet matematikai leírása mellett a dolgozat bemutatja az NMR spektroszkópiában elterjedt vektormodell egy olyan kibővített értelmezését, ami a csatolt rendszerek cserefolyamatait is leírja.

A kidolgozott elmélet alapján szimulációs programot fejlesztettem (ProMoCS: *Propagators & Monte Carlo for DNMR Spectrum Simulation*), amelynek előnye, hogy a szimuláció korábban teljesíthetetlenül nagy memóriaigénye különösebb elhanyagolások bevezetése nélkül is elfogadható mértékűre csökken. A szimuláció sebességének növelésére a programozástechnikai egyszerűsítések mellett a modern hardverek adta lehetőségeket is kihasználtam, így a program a többmagos processzorok vagy a több gépből álló hálózatok előnyeit is ki tudja használni. A hagyományosnak mondható paralellizálás mellett a legmodernebb párhuzamosítási technikát GPGPU videokártyák (*General Purpose computing on Graphics Processing Unit*) jelentik [22]. A program legfontosabb és legtöbb gépidőt igénylő eljárását a CUDA™ nyelvének (*Compute Unified Device Architecture*)

speciális igényeinek megfelelően újraírtam és sikeresen beprogramoztam. A kész programokat széleskörű teszteknek vettem alá hipotetikus rendszereken és valós példákön.

Az elméleti újításokat, a megírt program elméleti háttérét és alkalmazását korábban négy [1–4] cikkben közzétettük, ezeket foglalja össze ez a dolgozat. A 2. fejezet áttekinti a szimulációban használt Monte Carlo módszer, valamint a CUDA nyelvű programozás irodalmi háttérét és bemutatja ezen módszerek a dolgozat szempontjából fontos alapjait. A 3. fejezet a dinamikus NMR spektroszkópia irodalomból ismert elméletét foglalja össze, kitekintést nyújtva a spektrumszimuláció területére. A 4. fejezet PhD munkám elméleti kutatási eredményeit foglalja össze: bemutatja a matematikai formalizmust és értelmezi az eredményeket vektormodell kibővítése segítségével. Az 5. fejezetben bemutatásra kerül az elmélet alapján megírt program (ProMoCS), illetve ennek videokártyára implementált változata (ProMoCS-CUDA). Végül a 6. fejezetben néhány valós rendszer dinamikus NMR spektrumainak szimulációjával mutatom be a program teljesítőképességét.

2. fejezet

A szimulációs módszerek irodalmi háttere

2.1. Monte Carlo módszer

2.1.1. A legismertebb Monte Carlo szimulációk

A Monte Carlo szimulációk alatt olyan számolásokat értünk, amelyek egy rendszer átlagos tulajdonságát a rendszer néhány, véletlenszerűen választott tagjának viselkedésével írják le [23–28]. A szimuláció során mindig szerepet kapnak a véletlen számok. Ezek segítségével kerül kiválasztásra a makroszkopikus sokaság néhány tagja (ezt nevezik mintának). A minta elemeire meghatározzák a vizsgált mennyiség aktuális értékét (pl. belső energia, entrópia, feszültség stb.) és ezekből az aktuális értékekből számolható a makroszkopikus átlag, a szimuláció eredménye. A generált véletlen számok eloszlása, a minta kiválasztása, a mennyiség kiszámolása és a végeredmény meghatározása algoritmusonként változó, emiatt ez a módszert széles körben használható nagy szabadsági fokú rendszerek szimulációjára. A kémiában tipikus alkalmazási területük rendezetlen rendszerek szimulációja [29–33].

A kémiai alkalmazásokban legelterjedtebb az úgynevezett Metropolis Monte Carlo (MMC) [34], illetve az ennek mintájára megalkotott reverz Monte Carlo (RMC) módszer. A Metropolis Monte Carlo módszer legfontosabb mennyisége valamely U energijellegű potenciálfüggvény (pl. belső energia, szabadentalpia, szabadenergia, a fizikai modellől függően). A sokaság egy tagja $\min[1, \exp(-\Delta U / RT)]$ valószínűséggel kerül bele a mintába, ahol ΔU a vizsgált konfiguráció és a minta utolsó elemének potenciálkülönbsége. Ez a kiválasztási szabály biztosítja a minta Boltzmann-statisztika szerinti eloszlását. Erre a mintára ezután kiszámolják a vizsgált fizikai mennyiséget (pl. sűrűséget, felületi feszültséget, a geometriai paramétereket) és átlagolják a mintahalmazon. Ez a módszer önmagában, kísérleti adatok nélkül közelíti a vizsgált fizikai mennyiségeket, amelyet ezután kísérleti eredménnyel (pl. NMR spektrum) összevetve értelmeznek. Ilyen módszerrel vizsgálták például a különböző atomok eloszlását zeolitokban és más térhálós anyagokban [35–37]. Használják biopolimerek (fehérjék, DNS) szerkezetének felderítésére [38–41], adszorpció tanulmányozására [42–45], változó szerkezetű molekulák dinamikájának vizsgálatára [46, 47] és folyadékkristályok NMR spektroszkópiás szerkezetvizsgálatában [48–50] egyaránt.

A reverz Monte Carlo annyiban tér el a Metropolis Monte Carlo módszertől, hogy a minta elemeinek kiválasztását nem energijellegű potenciálfüggvény szabályozza. A spektroszkópiában például a kiválasztás a sokaság egy tagjának számított spektruma és a kísérleti spektrum közötti eltérésen alapul. Manapság az RMC standard mérésiértékelő mód-

szerré vált, gyakran használják diffrakciós mérések kiértékelésére [51–62], makromolekulák NMR-es szerkezetmeghatározására [63–67] és emellett spektrumfeldolgozásra írt algoritmusok alapját is képezi [68–71].

2.1.2. *A kinetikus Monte Carlo módszer*

A Metropolis Monte Carlo szimulációkban mindig egy egyensúlyi rendszer valamely jellemző paraméterét számolják ki. Ezek a szimulációk nem alkalmasak időbeli változások modellezésére. A Monte Carlo szimuláció tágabb értelemben mindazon számolásokat jelentheti, ahol a véletlen számok szerepet kapnak. Ezeknek egy csoportja, a kinetikus Monte Carlo módszerek (KMC), időbeli folyamatok modellezésére alkalmas.

A KMC alapvetően összetett rendszerek koncentrációviszonyainak statisztikus alapon történő becslésére vagy modellezésére használt módszer [72]. Determinisztikus módszerek erre a célra a változásokat leíró differenciálegyenlet-rendszert numerikusan oldják meg, így meghatározva a rendszer összetételét adott időpontokban. Ezzel szemben a kinetikus Monte Carlo egy megfelelő eloszlással generált véletlen szám segítségével határozza meg a következő változás (reakció) időpontját egy-egy speciesz esetében [73]. A végeredmény ebben az esetben is a rendszer összetételének meghatározása.

A KMC lényegesen különbözik az MMC és RMC szimulációktól abban, hogy nincs explicit kiválasztási szabálya. Az egyes reakciósorok megfelelő súlyozását, azaz megfelelő számú előfordulását, előállításuk menete biztosítja: a nagyobb valószínűségű reakcióutak többször kerülnek a mintába, mint a kevésbé valószínűk.

A KMC elmélete és matematikai alapjai körülbelül negyvenéves múltra tekintenek vissza, és azóta igen széles alkalmazási területei vannak [74–78]. A bonyolult reakciómechanizmusok feltárásán túl diffúziós és relaxációs folyamatok lefolyását is modellezzik ezzel a módszerrel. Az NMR spektroszkópia területén alkalmazták például DOSY spektrumok [79–82] illetve relaxációs modellek kiértékelésénél [83, 84].

A dolgozatban bemutatásra kerülő modell és az erre épülő program a kémiai cserefolyamatokat modellezi kinetikus Monte Carlo módszerrel. Lényeges különbség azonban a fent említettektől, hogy DNMR esetén a szimulált rendszer makroszkopikus egyensúlyban van, tehát nem az egyes komponensek koncentrációviszonyait vagy valamely időbeli változását akarjuk vizsgálni. Ebből a szempontból ennek a szimulációnak az MMC módszerrel rokon módon egy, az egyensúlyi rendszere jellemző mennyiség becslése a szimuláció célja.

2.2. A GPGPU alapú programozás

2.2.1. A CUDA alapjai

Az utóbbi években a számítógépes szimulációk térnyerése egyre fokozódik, köszönhetően az egyre gyorsabb processzorok megjelenésének. A többmagos processzorok megjelenésével lehetővé vált a számolási feladatok párhuzamosítása egyetlen PC-n belül is. Emellett megjelent a GPGPU (*General Purpose computing on Graphics Processing Unit*), amelyen a programok a grafikus kártyák adottságaira támaszkodva többszáz vagy több ezer szálon végzik a számításokat [85]. Ezekhez az általános GPGPU képes kártyákhoz a gyártók fejlesztői környezetet és nyelvi támogatást is adnak. Ilyen technológiával rendelkeznek az NVidia® CUDA™-képes kártyái [22], az AMD® FireStream™ processzorai [86] és az Apple® Mac bizonyos verziói (Snow Leopard™) [87]. A GPGPU programozása minden esetben a C nyelv megfelelő publikus bővítményével lehetséges. 2008-ban a GPGPU gyártók létrehoztak egy közös programnyelvet (OpenCL™), amely segítségével egységes nyelvi környezetben lehet a *kernel*függvényeket kódolni [88].

A GPGPU technológia lényege, hogy a párhuzamos programszálak (akár több ezer is lehet belőlük) különböző adatokon ugyanazt a műveletsort nagyon hatékonyan tudják elvégezni [89, 90]. Ezen tulajdonsága miatt ez a technológia kimondottan alkalmas mátrix- és vektorműveletek végzésére, valamint olyan szimulációs feladatok megoldására, amelyekben egy műveletsort több, egymástól független paraméterlistával kell kiszámolni. Az irodalomban fellelhető példák azt mutatják, hogy az ilyen számítások a GPGPU technológiával akár 50-200-szor gyorsabbak lehetnek, mint hagyományos CPU-n [22].

A videokártyákat elsősorban grafikai feladatok megoldására találták ki, de a GPGPU megjelenésével a természettudományokon belül számos területen hasznosítják ezek előnyeit [91–93]. Párhuzamosítható feladatok gyakran előfordulnak molekuladinamikai és Monte Carlo szimulációk során [94–99]. Írtak GPGPU programokat például kételektron-integrálok számítására [100, 101], elektronszerkezet számolására [102–104], reakciókinetikai problémák megoldására [105–107] és bioinformatikai feladatok megoldására [108–110].

A CUDA (*Compute Unified Device Architecture*) a C programnyelv kiterjesztése, amelynek segítségével GPGPU képes NVidia® videokártyák programozhatók. A hagyományos programozási technikáktól elsősorban egyedi párhuzamosítási lehetőségében és többszintű memóriakezelésében különbözik.

2.2.1.1. Programszálak

A CUDA alapelve röviden úgy fogalmazható meg, hogy “azonos műveletek, egyszerre, különböző adatokkal”. Az időigényes számításokat a kártyán sok, akár több ezer programszálon lehet végezni. A többszálúsítás akkor a leghatékonyabb, ha ezek a szálak ugyanazt a műveletet vagy műveletsort végzik különböző adatokkal, amely adatok a kártya saját memóriájában kerülnek tárolásra. A GPU a CPU-val együttműködik, a kétirányú adatforgalom biztosítja a jó és hatékony programozhatóságot [90, 111].

A programszálakat a CUDA-ban (legfeljebb háromdimenziós) blokkokba szervezik. A blokkok legfeljebb 512 szálat tartalmaznak. A gyakorlatban a blokkok számai nem teljesen egyszerre, hanem úgynevezett *warp*-okba rendezve hajtják végre az utasításokat. A blokkok méretét ezért úgy célszerű megválasztani, hogy az a *warp* méretének többszöröse legyen. A CUDA 1.3 verzióig a *warp* mérete 32 szál, az ideális blokkméret pedig 16×16 illetve 32×16 [111].

A szálakat tartalmazó egységes dimenziójú blokkokat a CUDA *grid*-be szervezi, így az egyszerre indítható szálak száma több ezer lehet. Az azonos *grid*-en lévő különböző blokkok szálainak végrehajtási sorrendje esetlegesen tekintendő, ezek nem tudnak egymással kommunikálni sem. Ez a szerkezet lehetővé teszi, hogy a végrehajtás során videokártyától függően a blokkok tetszőleges sorrendben akárhányszor is végrehajthatók legyenek [111].

A szálakat a blokkon belül, illetve a blokkokat a *grid*-en belül a CUDA automatikusan indexeli, így minden szál könnyen és egyértelműen azonosítható. A *grid*, illetve a blokkok dimenzióját a továbbiakban g , illetve d_x , d_y , a blokk *grid*-en belüli pozícióját b , a szálak blokkon belüli pozícióját pedig p_x , p_y jelöli.

Egy a *kernel*-en végrehajtandó eljárás meghívása során deklarálni kell a *grid* és a blokkok dimenzióit, a párhuzamosan futó szálak száma ezen méretek szorzata lesz: $g \cdot d_x \cdot d_y$. Ezek a szálak mindenképpen ugyanazt a számolási feladatot végzik el. Nem hatékony különböző feladatokat egy híváson belül a szálak között felosztani, mivel ebben az esetben a szálak a két feladatot egymás után hajtják végre, és nem egyszerre. Például, ha egy szálanként változó adat szerinti elágazás van a programkódban, akkor először az „igaz” ágat hajtják végre azok a szálak, amelyeknek azt kell, majd ezután a „hamis” ágat a többi. Emiatt például nem érdemes két tömb összeszorozásánál a nulla elemeket kiválogatni még akkor sem, ha ezek az adatok többségét kiteszik. A hagyományos programnyelveken megszokott egyszerűsítő elágazásoknak (pl. az említett nulla elemek szűrése) akkor van

értelme a CUDA-ban, ha az elágazási feltétel egy *warp* minden szálára egyforma eredményű.

A szálak kétszeres strukturálásának (*grid* és blokk) a memóriakezelésben (ld. 2.2.1.2. fejezet) és a szálak szinkronizálásban van szerepe. A *grid* különböző blokkokban lévő szálai a *kernel*en belülről nem szinkronizálhatók. Ezzel szemben az egy blokkban lévő a szálak szinkronizálhatók egymással (*syncthreads()* utasítás). A szinkronizálásnál figyelni kell arra, hogy az elágazások, illetve ciklusok belsejében kiadott *syncthreads()* utasítás a blokk minden szálára vonatkozik. A programkódot a processzorok addig hajtják végre az egyes szálakon amíg egy ilyen utasításhoz érnek, de nem ellenőrzik, hogy a kérdéses utasítás ugyanaz-e. Ha például egy elágazáson belül csak az egyik ágban van benne egy szinkronizálási lépés, akkor azok a szálak, amelyek a másik ágot hajtják végre, előre fognak „szaladni” a végrehajtásban. Általában tehát *syncthreads()* utasítást csak olyan helyre szabad tenni a programkódban, amit a blokk minden szála egyformán hajt végre.

2.2.1.2. Memóriakezelés

A CUDA szálkezelésének megfelelően különböző szintű memóriaterületeket definiál az egyes szálak, blokkok és a teljes *grid* szintjén. A szálak egyedi változóinak fenntartott helyek a 32 bites regiszterek. Ezek elérése gyors és a szálak a hozzáférés során nem ütközhetnek egymással. A regiszterek száma korlátozott, blokkonként legfeljebb 8192 (CUDA 1.1-ig), illetve 16384 (CUDA 1.2-től). Emellett a szálaknak lokális memóriaterülete is van, ennek mérete 16 kbyte. A regiszterek és a lokális memória között a fordító osztja el a szálak saját változóit.

Egy szálblokknak fenntartott memóriaterület a megosztott (*shared*) memória. Ez a memória gyorsan elérhető és a blokk minden egyes szála olvashatja és írhatja. Élettartama megegyezik a kernelhívás hosszával. A megosztott memóriaterületen keresztül bonyolítható adatforgalom, illetve adatmegosztás egy blokk különböző szálai között. A megosztott memória mérete, a ma megszokott méretekhez képest kicsi, néhány kbyte nagyságú [111].

Az eddigi memóriaterületeket kizárólag a kernel program érheti el, a *host* (azaz a CPU-n futó program) nem. A *grid* minden szála, valamint a *host* is eléri a globális (*global*) memóriaterületet. Ehhez a memóriához való hozzáférés lassabb, mint a megosztotthoz, ezért olyan adatokat, amelyekre a eljárás során többször szükség van, általában érdemes a megosztott memóriába másolni. Ez a memóriaterület használható változók közvetítésére a *host* és a *kernel* között, valamint a *kernel* életénél hosszabb ideig szükséges vagy a kisebb memóriaterületeken el nem férő adat tárolására.

A CUDA nyelvű programokban a többszintű memóriakezelés miatt a memóriaterületek optimális felhasználása és az írási-olvasási műveletek megfelelő koordinálása legalább olyan fontos, mint maga a programkód. Egy-egy eljárás lényegesen felgyorsítható azzal, ha a globális memóriából való olvasási és írási műveleteket a minimumra szorítjuk vissza és elkerüljük az olyan helyzeteket, amikor több szál ugyanahhoz az adathoz szeretne hozzáférni (*bank conflict*). A megosztott memóriaterülethez való hozzáférés optimalizálása is első-sorban a *bank conflict* megelőzését jelenti, azonban speciális olvasási műveletek miatt például „konfliktusmentes” az is, ha egy blokk minden szálának ugyanaz az adat kell.

2.2.2. Néhány példa a GPGPU alkalmazására

2.2.2.1. Egyszerű műveletek

A CUDA különösen hatékony mátrix- és vektorműveletek végzésére. Ez annak köszönhető, hogy az egy blokkon belüli programszálak többé-kevésbé szinkronizáltak, ha ugyanazokat a műveleteket kell elvégezniük. Ennek kihasználása az egyszerű mátrix- és vektorműveletekben a leglátványosabb.

Első példaként tekintsük egy mátrix másolását. Ez a feladat általában a memóriaterületek közötti adatmozgatás formájában jelentkezik, például az adatok betöltése a globális memóriából a megosztottba. A legegyszerűbb esetben a bemásolandó ζ mátrix dimenziója megegyezik a szálblokk dimenziójával, ami $d_x \times d_y$. Ekkor az Y eredménymátrix (p_x, p_y) indexű elemét, a szálblokk (p_x, p_y) indexű szála fogja betölteni és egy lépésben összesen $d_x \times d_y$ elemet másolnak át a szálak.

A szálblokk méreténél nagyobb mátrixok bemásolásánál minden egyes szála több elem másolása jut. Ekkor a másolást blokkonként érdemes elvégezni, azaz a (p_x, p_y) indexű szál a $(p_x + a \cdot d_x, p_y + b \cdot d_y)$ indexű elemeket tölti be, ahol a és b futóindexek.

Akkor is lehet blokkonként másolni, ha a bemásolandó mátrix mérete nem többszöröse a szálblokk méretének, csak az indexelési algoritmus bonyolultabb. Például egy vektor másolása esetén a szálblokkban minden egyes szálnak új indexet kell adni, például (p_x, p_y) helyű szál lineáris indexe $p_x \cdot d_x + p_y$ lehet. A másolás végén a blokk szálait minden esetben szinkronizálni kell [111].

A másoláshoz hasonlóan lehet gyorsan elemenkénti műveleteket végezni mátrixokkal vagy vektorokkal (pl. összeadás, skalárral szorzás). Valamivel bonyolultabb műveletet jelent egy tömbben lévő számok összegének meghatározása. Az „egyszálás” megoldás

ekkor a sorozatos összegzés. Több szálon ez a művelet úgy gyorsítható, hogy a szálak egy lépésben mindig két elemet adnak össze, majd a következő lépésben az előtte kapott eredmények közül kettőt és így tovább. Így minden lépésben feleződik az összeadandó tagok száma és n lépés után 2^n darab elem összegét kapjuk meg. Ebben az esetben minden egyes összeadási lépés után szinkronizálni kell a blokk szálait.

2.2.2.2. Nagy mátrixok szorzása

A mátrixok szorzása annyiban különbözik az egyszerű mátrixműveletektől, hogy egyetlen elem kiszámításához az eredeti mátrixok több elemét kell ismerni és az eredeti mátrixok elemei az eredménymátrix több elemét is befolyásolják.

Olyan mátrixok esetén, ahol a mátrix mérete kisebb, mint a CUDA szálblokk dimenziója (illetve a mátrixok beférnek a megosztott memóriába) a szorzáshoz a kiindulási mátrixokat (A és B) és az eredményt (C) a megosztott memóriában érdemes tárolni. A szorzat kiszámításánál minden egyes szál a C mátrix ráeső elemét számolja ki, azaz a (p_x, p_y) indexű szál C_{p_x, p_y} -t:

$$C_{p_x, p_y} = \sum_{a=1}^d A_{p_x, a} B_{a, p_y}, \quad (1)$$

ahol d az A mátrix oszlopainak száma (azaz B sorainak száma). Az a szerinti összegzés a programkódban egyetlen ciklusként jelenik meg, de nincs szükség az eredménymátrix elemein végigfutó másik két ciklusra. Ebben az esetben a szálblokk dimenziójának az eredménymátrix méreteivel kell megegyeznie.

A nagy mátrixok szorzási algoritmusát a kis mátrixokéra lehet felépíteni. Ebben az esetben a teljes *grid* ugyanazt az eredménymátrixot számolja, és minden egyes szálblokkra a C mátrix egy kisebb blokkjának kiszámolása jut (C_{sub}). Ezen blokkok dimenziója megegyezik a szálblokk dimenziójával és minden egyes szál ennek pontosan egy elemét számolja ki. A *grid* egyes blokkjai a C_{sub} mátrix kiszámítását blokkonkénti ciklusban végzik, ahol az A és B mátrixok egy-egy blokkjának szorzatát számolják ki egy lépésben, és ezalatt csak ezek a blokkok vannak a megosztott memóriába másolva. Minden egyes iteráció két lépésből áll: az A és B mátrixblokkok bemásolásából és ezek összeszorzásából. Mindkét lépés után szinkronizálni kell a szálakat majd az iterációs ciklus végén a blokk szálai az eredményül kapott C_{sub} mátrixot a globális memóriába másolják.

3. fejezet

Dinamikus NMR spektroszkópia

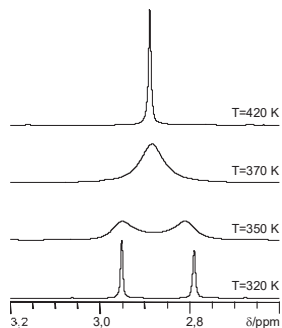
3.1. A dinamikus NMR jelenség

Az NMR spektrumok detektálási ideje többszáz ms. Ez alatt az idő alatt a minta atomi szinten olyan változásokon mehet keresztül, amelyek a detektált *fidet* és a spektrumot befolyásolják (pl. relaxációs folyamatok, kémiai cserefolyamat, diffúzió). Nagy általánosságban ezeket a jelenségeket tekintjük dinamikus jelenségeknek az NMR spektroszkópiában. A szűkebb értelemben vett dinamikus NMR spektroszkópia vizsgálatai olyan kémiai cserefolyamatokra korlátozódnak, amelyek időállandója az NMR időskálára esik. A cserefolyamat lehet valódi egyensúlyi reakció két különböző molekula között (intermolekuláris csere), vagy lehet egy molekula átrendeződése vagy konformációs mozgása is (intramolekuláris csere). Ez a „kémiai” különbség a reakció értelmezése szempontjából érdektelen. Ennél sokkal fontosabb különbség, hogy a folyamat két oldalán álló spinrendszerek egymással ekvivalensek (kölcsonös csere) vagy nem (nem kölcsonös csere).

A cserefolyamatok a hőmérséklet (reakciósebesség) függvényében jellegzetes NMR-jelalakváltozást okozhatnak: „alacsony” hőmérsékleten (a két jel frekvenciakülönbségénél és k^* -nál lényegesen lassabb reakciónál) az összes forma jelei jól láthatók, „magas” hőmérsékleten (a frekvenciakülönbségnél gyorsabb reakciónál) a jelek átlaga jelenik meg. A megjelenő jelek mindkét szélső tartományban élesek, a köztes tartományban szélesek [15]. A legszélesebb jeleket a koaleszcencia-hőmérsékleten tapasztaljuk. Ilyenkor a spektrum átmenetet képez az egy és a két csúcsból álló spektrum jelei között.

Az 1. ábrán látható a dimetil-acetamid ^1H NMR spektruma négy különböző hőmérsékleten. Ebben a molekulában az amidkötés körüli forgás szobahőmérsékleten gátolt, magasabb hőmérsékleten azonban a forgás szabaddá válik, így a két N-metil csoport megkülönböztethetetlen lesz. Ennek megfelelően 320 K hőmérsékleten (és az alatt) a spektrumban két éles jel van, amelyek a hőmérséklet emelésével kiszélesednek. Tovább emelve a hőmérsékletet a cserefolyamat gyorsul, a jelek összeolvadnak (370 K), majd ez a jel kiélesedik (420 K).

Amikor dinamikus NMR spektrumok szimulációjáról beszélünk, akkor az ilyen kinetikai folyamatoknak a spektrumok jeleire gyakorolt hatását írjuk le. Több hőmérsékleten vagy többfő-



1. Ábra Az N,N-dimetil-acetamid ^1H NMR spektruma négy különböző hőmérsékleten (250 MHz készülék, d_6 -DMSO).

le térérőnél felvéve a spektrumot a szimuláció segítségével meghatározhatók az érintett cserefolyamat aktiválási paraméterei is. A kapott adatok megbízhatósága nagymértékben függ a spektrumok minőségétől és a kiértékelés módjától is.

A legegyszerűbb összefüggés szerint, két szingulett jelet érintő egyetlen cserefolyamat (például a dimetil-acetamid amidkötés körüli rotációja) koaleszcencia állapotában a k_c sebessége együtthatót a következő összefüggés adja meg:

$$k_c = \frac{\pi}{\sqrt{2}} \Delta\nu, \quad (2)$$

ahol $\Delta\nu$ a két jel frekvenciájának különbsége [112].

Az 2. képlet legnagyobb hibája, hogy csak abban az esetben használható, ha mindkét jel szingulett, továbbá meg tudjuk állapítani a két elkülönülő jel eltolódását. A módszer további hibája, hogy a koaleszcencia hőmérsékletének meghatározása is pontatlan, mivel a jel legszélesebb állapotának környékén nagyon kicsi az eltérés két spektrum között. Többféle mágneses térben, vagy különböző magpárookra meghatározva a koaleszcencia-hőmérsékleteket, megkaphatók a cserefolyamat aktiválási paraméterei, de ezen adatok pontossága az említettek miatt nem elfogadható.

A pontosabb eredmények érdekében mindenképpen szükségessé válik a teljes spektrumsorozat szimulációja. Általában a paraméterek pontossága növelhető több cserélő magpár vagy nagyobb spinrendszerek szimulációjával. Ez utóbbinak előnye az egymagos szimulációkkal szemben, hogy a természetes jelszélesedés hatásai jobban kiszűrhetők, mivel ezek másképp hatnak a jelalakra, mint a cserefolyamat [12].

3.2. A dinamikus NMR jelenség értelmezése

3.2.1. Az egymagos vektormodell

Erős mágneses térben az $\frac{1}{2}$ spinű magoknak két megkülönböztethető spinállapota van, ezeket α és β állapotoknak nevezzük. A két állapot energiaszintje közötti különbség tipikusan MHz nagyságrendbe esik. A vektormodellben egy izolált mag spinállapotait vektorokkal szimbolizáljuk, amelyek hossza $\hbar\sqrt{I(I+1)}$ és mágneses térben úgy állnak be, hogy z irányú vetületük $I_z = m\hbar$ ($m = \pm\frac{1}{2}$). Az I operátor komponensei egymással nem kommutálnak, ezért a vektorok másik két komponense meghatározhatatlan, ezek az xy sík bármely irányába állhatnak, így az adott spinkvantumszámhoz tartozó vektorok egy forgáskúp felszínén helyezkednek el. Az α és β állapotú vektorok az energiakülönbségüknek

megfelelő frekvenciával precesszálnak a kúp tengelye körül. A vektorok eredője, a makroszkopikus mágnesezettség vektor (\mathcal{M}) egyensúlyban z irányú a két kúp eltérő populációja miatt.

A mérést indító 90° -os pulzust követően az \mathcal{M} vektor az xy síkba kerül, ez adja a detektálható jelet. A precesszió és a relaxációs folyamatok következtében az egyetlen kúp vektorai által adott jel (*fid*) exponenciálisan csökkenő amplitúdójú szinuszgörbe [15].

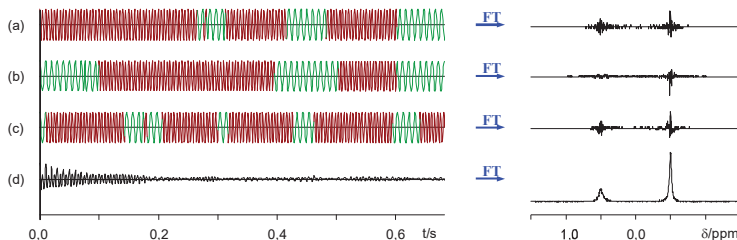
Többmagos spinrendszer esetén a vektormodell nem értelmezhető ilyen egyszerűen. Gyengén csatolt spinrendszerekre az egymagos spinrendszer modellje magonként alkalmazható. Így a vizsgált magtól különböző magokkal való csatolást a precessziós frekvenciába építhetjük, mivel a gyenge csatolás definíciója ($J_{\mu\mu'} \ll |\nu_\mu - \nu_{\mu'}|$) azt jelenti, hogy az egyes magok átmenetei csak elsőrendben függenek a többi mag spinállapotától. A detektált jel ebben az esetben az egyes magkörnyezetek által adott szinuszgörbék összege lesz.

A vektormodell eredeti formájában nem érvényes erősen csatolt spinrendszerekre, mivel ezeknél az egyes magok precessziója definíció szerint nem választható el egymástól.

3.2.2. A dinamikus NMR jelenség értelmezése csatolatlan spinekre

A dinamikus NMR jelenség leírására nagyon egyszerű modell ismert, amennyiben a jelenség egyetlen izolált spinhez kapcsolódik [15]. Ebben az esetben a detektált jelet (*fidet*) a 3.2.1. fejezetben ismertetett vektormodell makroszkopikus mágnesezettség vektora adja. Ez a vektor a detektálás ideje alatt a külső mágneses tér irányára merőleges xy síkban forog, frekvenciáját a mag kémiai környezete határozza meg.

A továbbiakban tekintsünk egy olyan magot, amely kétféle környezetben lehet, ezeket jelöljük $A^{(1)}$ -gyel és $A^{(2)}$ -vel. A mag kétféle környezetéhez tartozó két frekvenciája legyen $\omega^{(1)}$ és $\omega^{(2)}$. A kétféle környezet közötti átmenetet egy cserefolyamat biztosítja,



2. Ábra A dinamikus jelenség értelmezése egymagos spinrendszerben. (a–c) egy-egy mag *fid*-je és spektruma (d) száz ilyen *fid*, illetve spektrum átlaga.

amelynek sebességi együtthatói a két irányba k_{12} ill. k_{21} . A mérés során a rendszer makroszkopikus egyensúlyban van. A sebességi együtthatók alapján meghatározható a két állapot egyensúlyi relatív koncentrációja (K_1 és K_2):

$$K_1 = \frac{k_{21}}{k_{21} + k_{12}} \text{ illetve } K_2 = \frac{k_{12}}{k_{21} + k_{12}}. \quad (3)$$

A detektálás során a vizsgált mag K_1 , illetve K_2 valószínűséggel van az két konformáció egyikében. Ezen valószínűségek ismeretében egy pszeudo véletlen szám segítségével meghatározható az adott spinhalmaz konformációja a detektálás kezdetén (a példában $A^{(1)}$ lesz). A mágnesezetség vektor így a $\omega^{(1)}$ frekvenciával kezd körözni:

$$fid(t) = \exp(i\omega^{(1)}t). \quad (4)$$

A cserefolyamat egy idő után közbeszól, és a $t^{(1)}$ -gyel jelölt időpontban pillanatszerűen megváltoztatja a mag kémiai környezetét és így ebben az időpontban megváltozik a precessziós frekvencia is ($\omega^{(2)}$ lesz). Így detektált jel a $t \geq t^{(1)}$ időpontokra:

$$fid(t) = \exp(i\omega^{(2)}(t - t^{(1)}))\sigma^{(1)}, \quad (5)$$

ahol $\sigma^{(1)}$ a vektor fázisát adja meg a $t^{(1)}$ időpontban:

$$\sigma^{(1)} = \exp(it^{(1)}\omega^{(1)}). \quad (6)$$

Az 5. egyenlet egészen a következő csere időpontjáig ($t^{(2)}$) érvényes, ahol ugyanúgy frekvenciaváltás történik, mint az első cserénél, azaz a $t^{(2)}$ időpont után a detektált jelet a

$$fid(t) = \exp(i\omega^{(1)}(t - t^{(2)}))\sigma^{(2)} \quad (7)$$

összefüggés adja, ahol

$$\sigma^{(2)} = \exp(i\omega^{(2)}(t^{(2)} - t^{(1)}))\sigma^{(1)}. \quad (8)$$

Általánosan tehát az r . cserét követően a detektált jel:

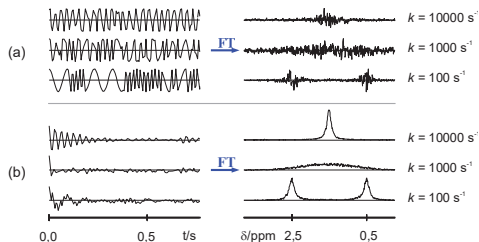
$$fid(t) = \exp(i\omega^{(1,2)}(t - t^{(r)}))\sigma^{(r)}, \quad (9)$$

ahol a $\omega^{(1,2)}$ értéke r paritásától függ.

Egy ilyen módon szimulált fid látható a 2.a ábrán. Ezt diszkrét Fourier-transzformálva megkapjuk a mag spektrumát. Ez a spektrum általában nagyon zajos, habár már sejteni lehet a jel alakját.

A cserék bekövetkezését az eredeti leírás szerint úgy lehet meghatározni, hogy megfelelően választott Δt időközönként egy ε véletlen számot generálunk, és akkor történik csere, ha ez a szám kisebb, mint $k_{12}\Delta t$. Ha k_{12} kicsi, a cserék ritkán következnek be, míg ha nagy, akkor sűrűn. A Δt időközöknek elég sűrűn kell lenniük a cseresebbséghez ké-

pest, különben torzítják az eredményt. Az első konformer véletlenszerű megválasztása és a cserék véletlenszerű időpontjai miatt különböző magok különböző *fidet* és más-más spektrumot eredményeznek (2.b–c ábra). A modellben néhány száz mag spektrumát összeadva az egyes magok spektrumának zaja kiátlagolódik (2.d ábra).



3. Ábra A sebességi együttható és a scanszám hatása a *fid* és a spektrum alakjára. (a) egyetlen mag szimulált *fidje* és spektruma, (b) 100 mag átlagaként kapott jel és spektrumok.

Az egyféle frekvenciával való precesszálás hossza az egyes magoknál más és más, de ezen időintervallumok átlaghossza a döntési kritériumon keresztül függ a sebességi együtthatóktól is. Lassú csere folyamat esetén az egyes frekvenciák külön érvényesülnek, így mindkét jelet detektáljuk. A sebességi együtthatók növelésével a különböző frekvenciájú részletek összekeverednek. Tovább növelve a cserék gyakoriságát, már csak a két jel átlaga figyelhető meg (3.a ábra), ami egyetlen jelet eredményez [15]. Több, azonos sebességi együtthatóval számolt *fidet* összeadva látható, hogy a lassú cserénél a *fid* két különböző szinuszgörbe szuperpozíciója, míg gyors cserénél egyetlen lecsengő szinuszgörbe (3.b ábra). A köztes sebességnél *fid*ek összhangja hamar megszűnik és ezért az összegük gyorsan lecseng, így a spektrumban kapott jel széles. A véletlen számok szerepe valamint a véletlenszerűen választott *fid*ek összegzése miatt ez a modell egy Monte Carlo szimulációs módszer alapjának tekinthető.

A dinamikus jelenség itt bemutatott értelmezése a makroszkopikus mágnesezettség vektor frekvenciaváltásán alapul, emiatt csak csatolatlan spinekre és olyan gyengén csatolt spinrendszerekben érvényes, ahol a cserepartnerek között nincs skaláris csatolás.

3.3. NMR spektrumok szimulációjának gyakorlata

3.3.1. Statikus NMR spektrumok szimulációja

3.3.1.1. Hamilton-operátor mátrixának felépítése

A dinamikus NMR spektrumok számolása a rendszer kölcsönhatási operátorainak felépítésére és ezek sajátértékproblémájának megoldására épül. A figyelembe vett köl-

csönhatások közül a legösszetettebb a spinek kölcsönhatása, azaz a kémiai eltolódás és a csatolási állandók rezonanciafrekvenciára gyakorolt hatása. Ezen kölcsönhatások leírása a dinamikus és a „statikus” (azaz kémiai cserefolyamatoktól mentes) rendszerekben azonos. Legyen n a spinhalmaz mérete, μ ill. μ' a spinrendszerben lévő magokat indexe, ν_μ a μ mag kémiai eltolódásának megfelelő frekvencia (Hz-ben), $J_{\mu\mu'}$ a μ és a μ' mag közötti csatolási állandó. Ekkor az oldatfázisú NMR statikus kölcsönhatásait leíró Hamilton-operátor frekvencia egységben a következő alakba írható [112]:

$$\hat{H} = 2\pi \sum_{\mu=1}^n \nu_\mu \hat{I}_{z\mu} + \pi \sum_{\mu=1}^n \sum_{\mu'=\mu+1}^n J_{\mu\mu'} (\hat{I}_\mu^+ \hat{I}_{\mu'}^- + \hat{I}_\mu^- \hat{I}_{\mu'}^+ + 2\hat{I}_{z\mu} \hat{I}_{z\mu'}), \quad (10)$$

ahol $\hat{I}_{z\mu}$ a μ mag spinoperátorának z komponense, \hat{I}_μ^+ és \hat{I}_μ^- pedig a μ mag léptető operátorai. Ezen operátorok a következőképpen hatnak egy $\frac{1}{2}$ spinű mag esetén:

$$\hat{I}_{z\mu} |\alpha_\mu\rangle = \frac{1}{2} |\alpha_\mu\rangle \quad \text{és} \quad \hat{I}_{z\mu} |\beta_\mu\rangle = -\frac{1}{2} |\beta_\mu\rangle, \quad (11)$$

valamint

$$\hat{I}_\mu^+ |\alpha_\mu\rangle = 0 \quad \text{és} \quad \hat{I}_\mu^+ |\beta_\mu\rangle = |\alpha_\mu\rangle, \quad (12)$$

illetve

$$\hat{I}_\mu^- |\alpha_\mu\rangle = |\beta_\mu\rangle \quad \text{és} \quad \hat{I}_\mu^- |\beta_\mu\rangle = 0. \quad (13)$$

A numerikus szimulációkban általánosan elterjedt ortonormált bázist a szorzatfüggvények alkotják (továbbiakban φ_a), amelyek a spinrendszerben lévő magok lehetséges spinfüggvényeinek szorzatai az összes lehetséges variációban. Ebben az esetben a mag μ indexét már nem szokás kitenni, azt a függvények sorrendje egyértelműen meghatározza. Ilyen szorzatfüggvényből n darab $\frac{1}{2}$ spinű mag esetén 2^n darab van.

A szorzatfüggvényekben az egyes magok operátorai csak az adott mag spinfüggvényére hatnak. A 11–13. egyenletek alapján a szorzatbázison a Hamilton-operátor mátrixának elemei a következők:

$$H_{ab} = \delta_{ab} \left(2\pi \sum_{\mu=1}^n m_\mu^{(a)} \nu_\mu + \frac{\pi}{2} \sum_{\mu=1}^n \sum_{\mu'=\mu+1}^n J_{\mu\mu'} D_{\mu\mu'}^{(a)} \right) + (1 - \delta_{ab}) \pi \sum_{\mu=1}^n \sum_{\mu'=\mu+1}^n J_{\mu\mu'} F_{\mu\mu'}^{(ab)}, \quad (14)$$

ahol $D_{\mu\mu'}^{(a)}$ értéke 1 illetve -1 , ha az a -edik bázisfüggvényben a μ -edik és a μ' -edik spinfüggvény paralel ill. antiparalel állású, $m_\mu^{(a)}$ az $\hat{I}_{z\mu}$ operátor φ_a bázisfüggvényhez tartozó sajátértéke és $F_{\mu\mu'}^{(ab)}$ értéke 1, ha az a -edik és a b -edik bázisfüggvények totálspinkvan-

tatszám azonos és pontosan az μ és a μ' magok spinfüggvényében különböznek, egyébként nulla [112].

A 14. egyenletben az első zárójelben szereplő tag a mátrix diagonális, míg a második annak offdiagonális elemeit tartalmazza. Látható, hogy a Hamilton-operátor mátrixa valós és szimmetrikus. A mátrix diagonalizálásával megkapjuk a rendszer sajátállapotait (ψ_k) és a hozzájuk tartozó energiákat (λ_k). A sajátállapotok felírhatók a szorzatfüggvények lineáris kombinációjaként:

$$\psi_k = \sum_a u_{ak} \varphi_a, \quad (15)$$

ahol az u_{ak} együtthatók valósak és a belőlük felépített U mátrix unitér.

A spinrendszer átmeneteinek frekvenciáját az energiaszintek különbsége adja meg. Az egyes frekvenciákhoz tartozó jelek intenzitásának számolásához be kell vezetni a teljes spinrendszer gerjesztő operátorát (\hat{I}^+):

$$\hat{I}^+ = \sum_{\mu=1}^n \hat{I}_{+\mu}. \quad (16)$$

A jelek intenzitása a gerjesztő operátor és azon két sajátállapot szorzatából számolható, amelyek közötti koherencia a jelet adja:

$$Y_{kl} = \left| \langle \psi_k | \hat{I}^+ | \psi_l \rangle \right|^2 = \left(\sum_{a,b} u_{ak} u_{bl} I_{ab} \right)^2, \quad (17)$$

ahol $I_{ab} = 1$, ha a $\varphi_a \rightarrow \varphi_b$ átmenet megengedett, és nulla egyébként [112].

3.3.1.2. Blokkosítás totálspinkvantszám szerint

A 14. egyenlet alapján a Hamilton-operátor mátrixának csak azon offdiagonális H_{ab} elemei lehetnek nullától különbözőek, ahol a φ_a és a φ_b bázisfüggvények totálspinkvantszám azonos (mivel az $F_{\mu\mu}^{(ab)}$ értéke csak így lehet 1). Emiatt a H mátrix totálspinkvantszám szerint blokkosítható. A totálspinkvantszám értéke n darab $\frac{1}{2}$ spinű mag esetén $n+1$ különböző értéket vehet fel, így a Hilbert-tér I_{tot} szerint $n+1$ invariáns alterére bomlik. A mátrix blokkdiagonális alakja miatt a Hamilton-operátor ψ_k sajátfüggvényei is az egyes alterekben lesznek. A H mátrix blokkjait, illetve az egyes altereket a továbbiakban $l=0..n$ indexeli, ahol az l indexű alterben lévő függvények totálspinkvantszám az $n/2 - l$.

A Hilbert-tér l indexű alterének dimenziója $\binom{n}{l}$. Az alter dimenziója az $l = n/2$ esetben lesz a legnagyobb, a mátrixblokk mérete ekkor lényegesen kisebb, mint a teljes mátrixé, így a szimulációban jelentős mennyiségű memóriaterület megspórolható [13]. A blokkdiagonális alakban a legnagyobb blokk mérete azonban ugyanúgy exponenciálisan nő, mint a teljes mátrixméret (2^n) , csak kisebb alappal.

3.3.2. Dinamikus NMR spektrumok szimulációja

3.3.2.1. Az átlag-sűrűségmátrix módszer

Az átlag-sűrűségmátrix módszerben a mozgásegyenlet a Liouville-von Neumann egyenleten alapul:

$$\frac{d\bar{\rho}}{dt} = -i[\bar{H}, \bar{\rho}]. \quad (18)$$

ahol $\bar{\rho}$ a spinrendszer átlag-sűrűségmátrixa, t az idő, \bar{H} pedig a rendszer Hamilton operátora. Ez utóbbit a 14. egyenlet alapján lehet felépíteni a spinrendszer összes magjának figyelembe vételével, beleértve a cserepartnerekben lévő spineket is.

A 18. egyenletet a φ_a bázisállapotok terében (Hilbert-térben) írtuk fel, hasonlóan a statikus NMR spektrumok szimulációjához. A kommutátor kezelése ebben a térben nehézkes, ezért első lépésként a 18. egyenletet az egykvantumátmenetek által kifeszített térben transzformálják (ami az átmenetek terének a szimulációban szerepet játszó M dimenziós altere, $M < 2^{2n}$ és a továbbiakban Liouville-térnek nevezzük). A transzformáció nyomán az átlag-sűrűségmátrix ($\bar{\rho}$) vektorrá alakul ($\bar{\sigma}$):

$$\bar{\sigma}_e = \bar{\rho}_{ab}. \quad (19)$$

A Hamilton-operátor szerepét átvevő operátor mátrixát \bar{L} jelöli és elemeit a következő összefüggés definiálja [113]:

$$\bar{L}_{ef} = \bar{H}_{ac}\delta_{bd} - \bar{H}_{ab}\delta_{ca}, \quad (20)$$

ahol e és f Liouville-térbeli indexek az (a,b) ill. (c,d) Hilbert-térbeli indexpárok megfelelői. Ekkor a Hamilton-operátorral vett kommutátor egy mátrixszorzássá egyszerűsödik:

$$[\bar{H}, \bar{\rho}] \rightarrow \bar{L}\bar{\sigma}, \quad (21)$$

és a 18. egyenlet alakja ebben a térben a következő:

$$\frac{d\bar{\sigma}}{dt} = -i\bar{L}\bar{\sigma}. \quad (22)$$

A 22. egyenlet a spinek egymással való kölcsönhatásait írja le. A dinamikus effektusok (relaxáció, cserefolyamatok) figyelembe vételéhez további tagokkal kell kiegészíteni az egyenletet. A cserefolyamatok leírásához definiálni kell minden cserefolyamat transzformációs mátrixát ($\Gamma^{(h)}$, h kivételesen a cserefolyamat indexe), amely a spinrendszer Hilbert-térbeli bázisfüggvényeit permutálja annak megfelelően, ahogy a cserefolyamat a magokat felcseréli. Ezen bázistranszformáció hatására a $\bar{\rho}$ mátrix új alakot ölt és a mozgásegyenletben az átlag-sűrűségmátrix ezen megváltozása szerepel:

$$\frac{d\bar{\rho}}{dt} = \sum_{h=1}^{m'} k_h \left(\Gamma^{(h)} \bar{\rho} \left(\Gamma^{(h)} \right)^T - \bar{\rho} \right). \quad (23)$$

ahol a k_h a h -adik cserefolyamat sebességi együtthatóját, m' a spinrendszerben definiált cserefolyamatok számát, $\left(\Gamma^{(h)} \right)^T$ pedig a $\Gamma^{(h)}$ transzponáltját (és egyben inverzét) jelöli.

A cserefolyamatokat a Liouville-térben leíró Kubo-Sack-féle [114, 115] mátrixot (\bar{X}) a 23. egyenletben definiált kifejezés transzformációjával kapjuk, így:

$$\bar{X}_{ef} = \sum_{h=1}^{m'} k_h \left(\Gamma_{ac}^{(h)} \Gamma_{bd}^{(h)} - \delta_{ac} \delta_{bd} \right), \quad (24)$$

ahol e és f Liouville-térbeli indexek az (a,b) ill. (c,d) Hilbert-térbeli indexpárok megfelelői.

A relaxációt egzaktul a Redfield-féle operátor írja le [116], amely a spinrendszer bármely két függvénye közötti relaxációt megengedi. Ehelyett a következő, a Bloch-egyenletekből származtatott egyszerűsített egyenletet is lehet használni [12]:

$$\frac{d\bar{\rho}}{dt} = \frac{\bar{\rho}_{eq} - \bar{\rho}}{T_1} - \frac{\bar{\rho}}{T_2}, \quad (25)$$

amely a Redfield-féle operátor diagonális közelítése (T_1 ill. T_2 a spin-rács, illetve a spin-spin relaxáció időállandója). Általában ez a tag tovább egyszerűsíthető, és a relaxáció operátora (\bar{R}) egy skalárrá egyszerűsödik:

$$\bar{R}_{ef} = -\delta_{ef} k^* = -\delta_{ef} \left(\frac{1}{T_1} + \frac{1}{T_2} \right), \quad (25)$$

amelynek a spektrumra gyakorolt hatása akár a szimuláció végén külön is számolható.

Látható, hogy az átmenetek terében történő szimuláció elsősorban a cserefolyamatokat leíró tag (23. egyenlet) miatt válik szükségessé.

A szimulációhoz megoldandó differenciálegyenlet a Liouville-térben:

$$\frac{d\bar{\sigma}}{dt} = -\left(i\bar{L} + \bar{R} + \bar{X} \right) \bar{\sigma}, \quad (27)$$

ahol a zárójelben szereplő \bar{L} , \bar{R} és \bar{X} valós szimmetrikus mátrixok, így a teljes mátrix komplex és szimmetrikus, tehát nem hermitikus. A mátrix sajátértékei emiatt komplex számok, amelyeknek képzetes része adja a jel frekvenciáját, valós része pedig a félértékszélességét [117]. A spektrum kiszámítására két lehetőség van: az $(i\bar{L} + \bar{R} + \bar{X})$ mátrix sajátértékproblémájának megoldása (hasonlóan a 3.3.1.1. fejezethez) vagy a 27. egyenlet, mint differenciálegyenlet megoldásán keresztül. Ez utóbbi közvetlenül a sűrűségmátrix időbeli változását adja meg:

$$\bar{\sigma}(t) = \exp(-i(\bar{L} + \bar{R} + \bar{X})t) \bar{\sigma}_0, \quad (28)$$

ahol $\bar{\sigma}_0$ az átlag-sűrűségmátrix a detektálás kezdetén ($t=0$). A detektált jel (*fid*) a sűrűségmátrix és az I^+ operátort reprezentáló sorvektor transzponáltjának szorzataként számolható ki [118]:

$$fid(t) = I^+ \bar{\sigma}(t). \quad (29)$$

Az 29. egyenlet alapján az ekvidisztáns mintavételi időpontokban kiszámolhatók a *fid* aktuális értékei, majd az így kapott időjel Fourier-transzformálásával megkapható a szimulált spektrum. Ez a megoldás nem feltétlenül igényli a sajátértékek kiszámítását és az első megoldással azonos eredményre vezet.

A H mátrixhoz hasonlóan az \bar{L} , \bar{R} és \bar{X} is blokkosítható totálspinkvantszám szerint. Az I^+ operátorral való szorzásban csak a sűrűségmátrix azon elemei érintettek, amelyek egykvantumátmenetnek tartoznak. Ezen elemek kiszámításához az $(i\bar{L} + \bar{R} + \bar{X})$ operátor egykvantumátmenetekre ható részét elég kiszámítani. Emellett a totálspinkvantszám szerinti blokkokat az \bar{L} mátrix örökli a H -tól és \bar{X} I^- -től, így ebben az esetben az átmenetek totálspinkvantszám-párjai szerint jönnek létre a blokkok. Az egykvantumátmeneteknél a nagyobb indexű átmenet egyértelműen megszabja a másik kvantszámát is. Ennek megfelelően ezek a blokkok a Hilbert-térbeliekhez hasonlóan indexelhetők ($l=1\dots n$). Az l indexű altérből az $l-1$ indexűbe történő átmenetek által alkotott altér dimenziója ekkor $\binom{n}{l-1} \binom{n}{l}$.

3.3.2.2. A dinamikus NMR spektrumokat szimuláló programok

Az első dinamikus spektrumokat számoló programokat az 1960-as évekre fejlesztették ki [113, 119–121]. Ezek a programok a 3.3.2.1. fejezetben ismertetett elven szimulálják a spektrumot. Az eredeti sorozat utolsó ismert változata a DNMR5 program volt [122], ami

az azóta született dinamikus NMR spektrumot szimuláló programok alapját képezi. Ezzel a programcsomaggal néhány (legfeljebb öt) csatoló mag spektrumát lehet kiszámolni úgy, hogy ezek között tetszőleges számú (elsősorú) cserefolyamat lehet.

Az utóbbi években kifejlesztett, új szemléletű programcsomag a MEXICO [123–126]. Ez a program is a 3.3.2.1. részben ismertetett elven számolja ki a spektrumokat, de jelentős programozástechnikai újításokat tartalmaz (ritka mátrixok hatékony tárolása [20], Lanczos-iteráció [21], dinamikus memória-allokáció), amelyeknek köszönhetően a régebbi DNMR5-nél lényegesen gyorsabb és stabilabb. A MEXICO egyelőre csak kétoldali cserefolyamat leírására alkalmas, és jelenleg publikált verziója csatolt spinek közötti nem-kölcsönös cserét rosszul szimulálja.

Szintén körülbelül öt magot tartalmazó spinrendszerek DNMR spektrumának szimulációjára és paramétereinek illesztésére alkalmas a WinDNMR 7.1 is [127–130]. Ez a program az előzőekhez hasonló elven működik, elterjedését jelentősen segítette a könnyen használható grafikus felülete.

A Bruker BioSpin® GmbH TopSpin™ programjának dinamikus NMR spektrumokat számoló TEDDY modulja [131] is a cserefolyamatokkal kiegészített Liouville-von Neumann-egyenlet (27. egyenlet) kvázistacionárius megoldásán alapul. A program tetszés szerinti spinrendszert (maximum 4–5 maggal), pszeudo-spinnel jellemzett spin-csoportokat [132] és ugyancsak tetszőleges számú, akár több magot is érintő cserefolyamatot tud kezelni.

Az említett programokon kívül a legtöbb spektrumfeldolgozó programcsomag tartalmaz dinamikus NMR spektrumok szimulációjára illetve spektrumparaméterek illesztésére alkalmas eljárásokat (MNova™-NMR [133], iNMR [134], SpinWorks [135], NUTS [136], NMRLoop [137]).

Speciális, kis spinrendszerekben a 27. egyenletben zárójelben szereplő mátrix analitikusan diagonalizálható, és ez alapján a spektrumparaméterek egyszerű nemlineáris paraméterillesztéssel is megkaphatók (pl. 4.5.2.1. leírtak alapján vagy Ref. [118]-ben).

4. fejezet

Elméleti eredmények

4.1. Alapfogalmak

4.1.1. A spinhalmaz

A szimuláció kulcsfogalma a spinhalmaz, ami olyan S_1, S_2, \dots, S_n atommagok rendezett halmaza, amelyek a spinhalmaz elemein kívül más magokkal nem csatolnak. Dinamikus jelenség esetén a spinhalmaz többféle kémiai környezetben fordul elő, ezeket a továbbiakban konformereknek nevezzük és $A^{(1)}, A^{(2)} \dots$

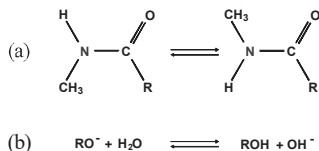
$A^{(m)}$ -mel jelöljük (m a konformerek szá-

ma). Ez a fogalom intramolekuláris cserefolyamatokban a hagyományos értelemben vett konformereket jelenti (4.a ábra), intermolekuláris cserefolyamatokban pedig a reakcióegyenlet egyik oldalán lévő spinhalmaz kémiai környezetét írja le (4.b ábra). A konformerek közötti átmenetek során csak a spinhalmaz környezetét leíró spektroszkópiai paraméterek (kémiai eltolódás, csatolási állandó) változnak meg (a cserefolyamat egy hasonló értelmezése található [138]-ben is).

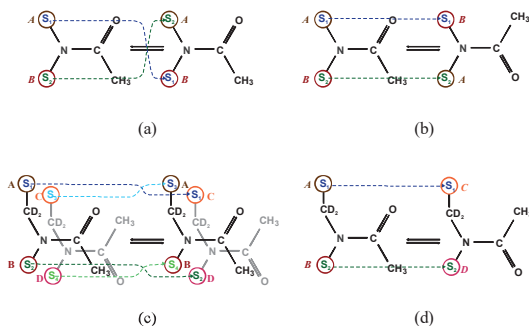
A 3.3.2.1. fejezetben az összes operátor mátrixát a teljes spinrendszer bázisfüggvényein írtuk fel, az átlag-sűrűségmátrixon alapuló szimulációkban erre van szükség. A spinrendszer az összes magot tartalmazza, amely valamelyik spinhalmazban szerepel. Tehát a spinrendszer nem csak a J -csatolásra, hanem a cserefolyamatokra nézve is zárt halmaz és általában nagyobb, mint egy spinhalmaz.

A két definíció és a cserefolyamat kapcsolatát a 5. ábrán hasonlítjuk össze. A spinrendszerben a spineket mágneses tulajdonságuk és kémiai környezetük alapján különböztettük meg, a magok sorrendje a cserefolyamatok során megváltozik. Ezt szemlélteti az 5.a ábra egy kölcsönös csere esetén, ahol S_1 ^1H mag az A környezetből B-be kerül, így S_2 lesz belőle, míg az eredeti S_2 mag eközben a B-ből az A-ba jutva S_1 lesz.

A spinhalmazban ezzel szemben az egyes spineket a halmazban elfoglalt helyük definiálja. Ez az atomi sorrend a cserefolyamatok során nem változik meg, a kémiai környezet leírása pedig csak a konformerekben jelenik meg. Az 5.b ábrán látható, hogy az S_1 mag mindvégig az első helyen marad, de környezete A vagy B is lehet. Eközben az S_2 mag is megőrzi helyét, csak a környezete vált B-ből A-ba, és vissza. A magok szempontjából a



4. Ábra A konformerek értelmezésének lehetőségei. (a) Kémiai értelemben vett konformerek (rotamerek) cseréje (az ^1H spinhalmaz mérete: $n = 4$ mag). (b) Kémiai reakciók, mint cserefolyamatok (az ^1H spinhalmaz mérete: $n = 2$ mag).



5. Ábra A spinhalmaz valamint a spinrendszer és a cserefolyamat kapcsolata. S-ek a spinhalmaz magjai, A, B, C és D ezek környezeteit jelöli. (a) Kölcsönös csere értelmezése spinrendszerrel. A és B környezet az acetamid két hidrogénjének környezetét jelöli. (b) Ugyanaz, mint (a), de spinhalmazzal. (c) *N*-(dideutero-metil)-acetamid NH és NCD₂H helyzetű ¹H magjainak nem-kölcsönös cseréjének értelmezése spinrendszerrel. (d) Ugyanaz, mint (c), de spinhalmazzal.

csere előtti és utáni állapot két különböző konformert jelent, annak ellenére, hogy ekvivalensek.

A kétféle értelmezés között még látványosabb a különbség nem kölcsönös csere esetén. Az 5.c ábrán az $AB \leftrightarrow CD$ cserefolyamat spinrendszere négy ¹H magot tartalmaz. Ezek mindegyikének definiálásához a „reakcióegyenlet” egy-egy oldalán meg kell duplázni a „molekulákat”. Miközben az S₁ és S₂ magok az AB, addig S₃ és S₄ magok a CD környezetben találhatók. A csere során az S₁ az S₃-mal, míg S₂ az S₄ maggal cserél helyet.

Ezzel szemben a spinhalmazon alapuló értelmezésben (5.d ábra) az S₁-S₂ magpár vagy az AB, vagy a CD környezetben van, és a párok *a priori* állandók. A cserefolyamatok során a spinek atomi sorrendje továbbra sem változik. A legfontosabb különbség a spinrendszeren alapuló leírástól az, hogy ebben a modellben nincs szükség a reakcióegyenlet meg többszörözésére.

Látható, hogy a spinrendszerben a skalárcsatolást és a magcserét szimultán kölcsönhatásként kell kezelni, ezért azok nem szeparálhatók, így leírásukhoz nagyobb atomi bázis kell. Ezzel szemben, a spinhalmazban külön kezelhető a kétféle kölcsönhatás, a spinrendszer így kisebb darabokra bomlik.

4.1.2. Az egyedi sűrűségmátrix

A 3.3.2. fejezetben a szimulációs gyakorlatban elterjedt sűrűségmátrixot ($\bar{\rho}$ ill. $\bar{\sigma}$) következetesen *átlag-sűrűségmátrix*nak neveztük. Ez a mátrix a spinrendszer átlagos állapotát adja meg és ebből az átlagból lehet kiszámolni a makroszkopikus rendszer egészére jellemző spektrumot.

Ezzel szemben a továbbiakban bemutatásra kerülő elméletben a spinhalmazok *egyedi* viselkedését modellezzük. Ennek érdekében definiáljuk a spinhalmazok *egyedi* sűrűségmátrixát (ρ vagy σ). Az egyedi sűrűségmátrix – az átlagoshoz hasonlóan – a pulzusok, a precesszió és a cserefolyamatok hatására változik, e hatásokat figyelembe véve bármely időpontban egzaktul kiszámolható. Azonban, míg az átlag-sűrűségmátrix a *spinrendszer* adott időpontban lehetséges állapotainak *átlaga*, addig az egyedi sűrűségmátrix a *spinhalmaz egy* lehetséges állapotát tartalmazza.

Az egyedi sűrűségmátrixok alapvetően azért különböznek egymástól, mert az egyes spinhalmazoknál a cserék más-más időpontokban következnek be. Két szomszédos cserepont közötti intervallumot a továbbiakban időszeletnek nevezzük. Egy ilyen időszeleten belül a spinhalmaz konformációja állandó. Egy adott spinhalmazhoz rendelt időszeletek és az ezekben szekvenciálisan megvalósuló konformerek együttesét nevezzük a spinhalmaz életútjának (4.2. fejezet). Így tulajdonképpen a spinhalmazokat és az egyedi sűrűségmátrixukat életútjuk, azaz a konformációváltozások (cserék) időpontjai és termékei különböztetik meg egymástól. Az egyedi sűrűségmátrix így azt azon állapotok átlagát tartalmazza, amelyek életútja azonos. Ezek az állapotok a kvantummechanika törvényei szerint megkülönböztethetetlenek, tehát az átlag makroszkopikusan egyedinek tekinthető. Ugyanakkor az átlag-sűrűségmátrix elméletileg megkülönböztethető állapotokat átlagol, mivel a cserefolyamatok kinetikája leírható klasszikus módszerekkel.

A spinhalmaz életútjának ismeretében tetszőleges időpontban előállíthatjuk az egyedi sűrűségmátrixát. A számolás kulcslépése az időben propagáló mátrix meghatározása (4.3. fejezet). Az egyedi sűrűségmátrix változásának ismeretében kiszámolható a spinhalmazhoz tartozó, a teljes életút alatt mintavételezett *fid* és spektrum, amit egy *scannek* nevezünk.

A Monte Carlo szimulációban néhány száz (100 – 2000) *scant* számolunk ki, ezek összege adja a szimulált spektrumot. A spinhalmazok egyedi életútja miatt a *scanek* különbözőek lesznek, ez biztosítja a szimulációhoz megkövetelt sokaság változatosságát.

4.2. A cserefolyamatok kinetikai leírása

Legyen a reakciók általános egyenlete:



ahol $A^{(h)}$ ill. $A^{(g)}$ közt az m -féle lehetséges konformerből, és az ezek közötti cserefolyamat látszólagos sebességi együtthatója k_{gh} . Tegyük fel, hogy minden reakció pszeudo elsőrendű. A makroszkopikus egyensúly feltételezése lehetővé teszi a nem valódi elsőrendű cserefolyamatok kezelését is (például 4.b ábra), mivel így az állandó koncentrációk biztosítják a látszólagos sebességi együtthatók változatlanóságát.

Ekkor a rendszer kinetikai viselkedését leíró egyenletrendszer j -edik egyenletének általános alakja:

$$\frac{d[A^{(h)}]}{dt} = \sum_{g=1}^m (-k_{hg}[A^{(h)}] + k_{gh}[A^{(g)}]), \quad (31)$$

ahol a nem létező reakciókra $k_{hg} = 0$, és h , illetve g befutja az összes indexet ($1 \dots m$).

Feltételeztük, hogy a rendszer a cserefolyamatok szempontjából egyensúlyban van, azaz minden h indexre teljesül, hogy

$$\frac{d[A^{(h)}]}{dt} = 0. \quad (32)$$

A további leírás megkönnyítése érdekében bevezetjük az $A^{(h)}$ konformerek relatív gyakoriságát (K_h):

$$K_h = \frac{[A^{(h)}]}{\sum_{g=1}^m [A^{(g)}]}. \quad (33)$$

Ha a 32. egyenletbe behelyettesítjük a 31. egyenlet jobb oldalát minden egyes j -re és a koncentrációkról áttérünk a relatív gyakoriságokra, egy lineáris egyenletrendszert kapunk, amelynek h -adik egyenlete a következő:

$$0 = \sum_{g \neq h} (-k_{hg}K_h + k_{gh}K_g). \quad (34)$$

Ezt az egyenletrendszert kiegészítve a

$$\sum_h K_h = 1 \quad (35)$$

egyenlettel, a 34–35. egyenletrendszer megoldása megadja az egyes konformerek relatív gyakoriságát a rendszerben.

A spinhalmazok életútjának meghatározása két részre oszlik: a cserék időpontjainak $(t^{(r)})$ és az egyes időszelletekben jelen lévő konformerek $(A^{(r)})$ meghatározására. Ez utóbbi lépés az első időszeletre más, mint az összes azt követőre.

A detektálás kezdetén ($t = 0$ időpontban) a szimulált spinhalmaz konformációját véletlenszerűen választjuk meg úgy, hogy a valószínűségi változó eloszlása megegyezzen a K_h relatív gyakoriságokkal.

A csereidőpontok megfelelő eloszlásának biztosításához először meg kell határozni az egyes konformerek élettartamát. Mivel minden reakció elsőrendű, bármely $A^{(h)}$ speciesz bomlásának kinetikája is elsőrendű, amelynek sebességi együtthatója (d_h) a 31. egyenlet alapján:

$$d_h = \sum_{g=1}^m k_{hg} . \quad (36)$$

ahol m a konformerek száma és $k_{hh} = 0$. Annak a valószínűsége $(V_{\Delta t}^{(h)})$, hogy a spinhalmaz Δt ideig az $A^{(h)}$ konformációban marad a következő:

$$V_{\Delta t}^{(h)} = d_h \exp(-d_h \cdot \Delta t), \quad (37)$$

tehát a csere nélküli időintervallumok hossza exponenciális eloszlású. Így az r . csere időpontja $(t^{(r)})$ az $r-1$. $(t^{(r-1)})$ ismeretében a következő rekurzív képlettel adható meg:

$$t^{(r)} = t^{(r-1)} - \ln(\varepsilon) / d_h, \quad (38)$$

ahol ε egy egyenletes eloszlású véletlen szám 0 és 1 között, így a 37. egyenletben az exponenciális előtt szereplő d_h faktorról nem kell ε -t leosztani. A $t^{(r-1)}$ és $t^{(r)}$ közötti időintervallumot nevezzük a továbbiakban az r -edik időszelethez, amelynek hossza $\Delta t^{(r)} = t^{(r)} - t^{(r-1)}$.

Ha a spinhalmaznak kettőnél több lehetséges konformere van ($m \geq 3$), akkor bomlási időpontok mellett minden egyes cserénél ki kell választani a bomlásterméket is. Ennek megválasztása véletlenszerűen történik, úgy, hogy annak a valószínűsége, hogy $A^{(h)}$ -ból $A^{(g)}$ keletkezik (κ_{hg}) :

$$\kappa_{hg} = k_{hg} / d_h = k_{hg} / \sum_{l=1}^m k_{hl} . \quad (39)$$

A κ_{hg} mennyiségekre teljesül a valószínűségi függvény három kritériuma és a termékek κ_{hg} szerinti választása biztosítja a párhuzamos reakciók elméletének megfelelő eloszlást.

A 38. egyenlet alapján meghatározott $t^{(1)}$, $t^{(2)}$, stb. időpontok és az egyes időszeletekben jellemző konformációk (39. egyenlet) egyértelműen megadják a spinhalmaz életútját, azaz leírják viselkedését a teljes detektálási idő során.

A szimulációk során a cserefolyamatok hőmérsékletfüggő sebességi együtthatóit az Eyring-Polányi-egyenlet alapján az aktiválási paraméterekből határozzuk meg:

$$k_{hg} = \frac{\kappa \cdot k_B}{h_p} \cdot T \cdot \exp\left(-\frac{\Delta H_{hg}^{++} - T\Delta S_{hg}^{++}}{RT}\right), \quad (40)$$

ahol h_p , k_B és R a Planck, Boltzmann és az egyetemes gázállandó, κ a transzmissziós koeficiens, ΔH_{hg}^{++} és ΔS_{hg}^{++} a reakció aktiválási entalpiája és entrópiája és T a hőmérséklet.

4.3. A *fid* szimulációja

4.3.1. Precesszió (propagálás)

A spinhalmaz kémiai környezetét leíró paraméterek a különböző konformerek esetén eltérők. Emiatt az egyes konformerek Hamilton-operátorai is különbözők lesznek, a 14. egyenlet alapján:

$$H_{ab}^{(h)} = \delta_{ab} \pi \left(2 \sum_{\mu=1}^n m_{\mu}^{(a)} v_{\mu}^{(h)} + \frac{1}{2} \sum_{\mu=1}^n \sum_{\mu'=\mu+1}^n J_{\mu\mu'}^{(h)} D_{\mu\mu'}^{(a)} \right) + (1 - \delta_{ab}) \pi \sum_{\mu=1}^n \sum_{\mu'=\mu+1}^n J_{\mu\mu'}^{(h)} F_{\mu\mu'}^{(ab)}, \quad (41)$$

ahol az változók általában megegyeznek a 10. és a 14. egyenletben használtakkal, a h felső index pedig a konformer sorszáma.

A további számolások során a Hamilton-operátor konformertől való függése az időszeletektől való függés formájában jelenik meg. Így a 41. egyenletben a h index $h^{(r)}$ -re módosulna, amit az átláthatóság kedvéért röviden r indexszel jelölünk, így a jelölés $H^{(r)}$ -re módosul. Hasonlóan változik a Hamilton-operátor mátrixától függő változók jelölése: $A^{(r)}$ a sajátértékekből álló diagonális mátrixot, $U^{(r)}$ a sajátvektorokból álló unitér mátrixot jelöli, azaz:

$$H^{(r)} = U^{(r)} A^{(r)} (U^{(r)})^T. \quad (42)$$

A *fid* számolásához szükséges képleteket először a Liouville-térben vezetjük le. Ennek érdekében a 20. egyenlethez hasonlóan definiáljuk az egyes konformerek $L^{(r)}$ mátrixát:

$$L_{ef}^{(r)} = H_{ac}^{(r)} \delta_{bd} - H_{db}^{(r)} \delta_{ca}, \quad (43)$$

ahol e és f Liouville-térbeli indexek az (a, b) ill. (c, d) Hilbert-térbeli indexpárok megfelelői. Az $L^{(r)}$ mátrixot a $C^{(r)}$ unitér mátrix diagonalizálja, a sajátértékeit a $\Omega^{(r)}$ diagonális mátrix tartalmazza, azaz:

$$L^{(r)} = C^{(r)} \Omega^{(r)} (C^{(r)})^T. \quad (44)$$

A precesszió vagy propagálás a magára hagyott spinhalmaz sűrűségmátrixának változását írja le olyan időintervallumban, ahol nem történik magcsere. Ilyen peremfeltételekkel a propagálás operátora $(P^{(r)})$ konformer-, illetve időszéletfüggő, viszont a cserefolyamatok sebességi együtthatóitól független (ellentétben a 28. egyenletben szereplő exponenciális operátorral).

A Δt idővel történő propagálás az r -edik időszéletben alábbiakban definiált $P^{(r)}(\Delta t)$ operátorral történik [123]:

$$P^{(r)}(\Delta t) = \exp(iL^{(r)}\Delta t), \quad (45)$$

ahol $L^{(r)}$ a 43. egyenletben definiált konformerfüggő mátrix. A továbbiakban a propagálást mindig az aktuális (r -edik) időszélet elejétől $(t^{(r-1)})$ végezzük, azaz $\Delta t = t - t^{(r-1)}$. A 45. egyenletben definiált $P^{(r)}$ propagátor mátrixának kiszámolásához az $L^{(r)}$ valós szimmetrikus mátrixot diagonalizáljuk a 44. egyenlet szerint. Az r -edik időszéletben lévő t időpontban (t a detektálás kezdete óta eltelt idő) a $\sigma(t)$ időfüggő sűrűségmátrix úgy számolható ki, hogy az időszélet eleji vektort $(\sigma^{(r-1)})$ megszorozzuk a 45. egyenletben szereplő operátor 44. egyenlet alapján kifejtett alakjával:

$$\sigma(t) = \exp(iL^{(r)}\Delta t) \sigma^{(r-1)} = C^{(r)} \exp(i\Omega^{(r)}\Delta t) (C^{(r)})^T \sigma^{(r-1)}. \quad (46)$$

A 46. egyenlet első felében szereplő exponenciális kifejezést sokféleképpen ki lehet számolni. A későbbiekben látni fogjuk, hogy a szimulációhoz és a képletek értelmezéséhez is a diagonalizálásra épülő megoldás a célravezető, ezért a továbbiakban ez alapján a formalizmus alapján írjuk a képleteket.

4.3.2. Cserefolyamatok

A cseréről feltételezzük, hogy pillanatszerűek, ezért a Φ_e bázisfüggvények relatív mágnesezettsége – azaz a sűrűségmátrix elemei – csere közben nem változnak meg. A csere során az aktuális konformer változik meg, azaz az $L^{(r)}$ operátor és annak sajátfüggvényei $(\psi_p^{(r)})$, valamint sajátértékei $(\omega_p^{(r)})$. Ezáltal a propagálás operátora is megváltozik, és

az új propagátor csak az új időszület elejétől érvényes. Emiatt a további számolásokhoz a csere időpontjában ($t^{(r)}$) ki kell számolni a pillanatnyi sűrűségmátrixot.

A σ mátrixot $t^{(r-1)}$ -ből a $t^{(r)}$ időpontba propagáló összefüggés megkapható a 46. egyenletben impliciten szereplő t változó $t^{(r)}$ -re való cseréjével (azaz Δt kifejezést $\Delta t^{(r)}$ -re cseréljük):

$$\sigma^{(r)} = C^{(r)} \exp(i\Omega^{(r)}\Delta t^{(r)}) \left(C^{(r)}\right)^T \sigma^{(r-1)}. \quad (47)$$

A 47. egyenlet rekurzívan használható, segítségével minden egyes csereidőpontban (azaz minden időszület elején) megkapjuk a sűrűségmátrixot, ha ismerjük az értékét a detektálás kezdetén ($\sigma^{(0)}$). Az egypulzusos kísérletben a 90° -os pulzus után ($t = 0$) csak transzverzális mágnesezettség van jelen. Ebben az állapotban a sűrűségmátrix arányos a gerjesztő operátort a Φ_e bázisfüggvények bázisán reprezentáló sorvektor (I^+) transzponáltjával, azaz $\sigma_e^{(0)} = 1$, ha a Φ_e egykvantumátmenet megengedett, egyébként nulla.

4.3.3. Detektálás

A t időpillanatban az adott molekula által adott jelet a sűrűségmátrixának ($\sigma(t)$) és a gerjesztő operátor sormátrixának (I^+) szorzata adja (formailag hasonlóan a 29. egyenlet-hez, de ott az operátorok mátrixa más bázison van felírva):

$$fid(t) = I^+ \sigma(t). \quad (48)$$

A t időpontbeli sűrűségmátrix a 46. egyenlet alapján számolható $\sigma^{(r-1)}$ -ből (ahol t az r -edik időszületben van, azaz $t^{(r-1)} \leq t < t^{(r)}$), így a t időpontban detektált jel:

$$fid^{(r)}(t) = I^+ C^{(r)} \exp(i\Omega^{(r)}\Delta t) \left(C^{(r)}\right)^T \sigma^{(r-1)}. \quad (49)$$

ahol $\sigma^{(r-1)}$ helyébe mindig annak az időszületnek az elején érvényes sűrűségmátrixot kell behelyettesíteni, amelyikbe a detektálás időpontja esik. Az 49. egyenlet első két tényezőjének szorzatát $Q^{(r)}$ -rel jelölve:

$$fid^{(r)}(t) = Q^{(r)} \exp(i\Omega^{(r)}\Delta t) \left(C^{(r)}\right)^T \sigma^{(r-1)}. \quad (50)$$

Az $Q^{(r)}$ sormátrix a gerjesztő operátor transzponáltjának mátrixa a sajátfüggvények bázisán, amelynek elemei az egyes átmenetek amplitúdói.

Az 50. egyenlet olyan t értékekre adja meg a fid értékét, amelyek az adott időszületen belülre esnek. A $t < t^{(r-1)}$ és a $t \geq t^{(r)}$ esetekben a $fid^{(r)}(t)$ függvény nincs értelmezve.

Egyetlen *scan* *fid*-je (*FID*) az egyes időszületekre számolt *fid*-ek uniója (\cup):

$$FID(t) = \bigcup_r fid(t) = \bigcup_r Q^{(r)} \exp(i\Omega^{(r)} \Delta t) \left(C^{(r)}\right)^T \sigma^{(r-1)}, \quad (51)$$

ahol az időszeletek fid -jeinek unióját úgy értelmezzük, hogy az eredmény értelmezve van az összes $fid^{(r)}$ függvény értelmezési tartományának unióján és értéke minden t időpontban megegyezik azon egyetlen $fid^{(r)}$ függvény t -beli értékével, ami t -ben értelmezve van. Ez a definíció a egyértelműen definiálja $FID(t)$ értékét, mivel a $fid^{(r)}$ függvények értelmezési tartományai páronként diszjunktak és együtt a teljes időskálát lefedik.

A teljes $fid(F)$ az egyedi *scanek* összegeként számolható ki:

$$F(t) = \sum_s FID(t) = \sum_s \bigcup_r Q^{(r)} \exp(i\Omega^{(r)} \Delta t) \left(C^{(r)}\right)^T \sigma^{(r-1)}. \quad (52)$$

A szimuláció során az $F(t)$ függvény értékét diszkrét pontokban számoljuk ki egy véges időintervallumon, majd ezt a tömböt diszkrét komplex Fourier-transzformálva kapjuk a szimulált spektrumot.

4.3.4. A spektrum direkt szimulációja

A spektrum kiszámolható az F függvény diszkrét pontokra bontása nélkül is. Az időszület fid -jét megszorozva a relaxációt helyettesítő $\exp(-k^*t)$ taggal (k^* a relaxáció látszólagos időállandójának reciproka) és a függvényt folytonos Fourier transzformálva kapjuk az r -edik időszület elméleti spektrumát ($Y^{(r)}(\omega)$):

$$Y^{(r)}(\omega) = \int_{t^{(r-1)}}^{t^{(r)}} fid^{(r)}(t) \exp(-i\omega t) \exp(-k^*t) dt, \quad (53)$$

A $fid^{(r)}$ korábban levezetett alakját (50. egyenlet) behelyettesítve és a 46. egyenlet első, egyszerűbb alakját használva:

$$Y^{(r)}(\omega) = \int_{t^{(r-1)}}^{t^{(r)}} I^+ \exp(-iL^{(r)} \Delta t) \sigma^{(r-1)} \exp(-(i\omega + k^*)t) dt, \quad (54)$$

Az 54. egyenletet átalakítva ($L^{(r)} = L^{(r)} - \omega$):

$$Y^{(r)}(\omega) = I^+ \left(\int_0^{\Delta t^{(r)}} \exp((iL^{(r)} - k^*)\Delta t) d\Delta t \right) \exp(-(i\omega + k^*)t^{(r-1)}) \sigma^{(r-1)}, \quad (55)$$

majd az 55. egyenletet kiintegrálva és összegezve minden egyes időszületre (r) és *scanre* (s), a kapott görbe valós része lesz a szimulált abszorpciós NMR spektrum:

$$Y(\omega) = I^+ \sum_s \sum_r \frac{1 - \exp((iL^{(r)} - k^*)\Delta t^{(r)})}{k^* - iL^{(r)}} \exp(-(i\omega + k^*)t^{(r-1)}) \sigma^{(r-1)}. \quad (56)$$

Az 56. egyenlet szerinti Monte Carlo szimuláció még $L^{(r)}$ diagonális alakjának használatával is nagyon időigényes lenne, mivel a meglehetősen összetett számolást el kell végezni minden egyes (s, r) indexpárra minden frekvenciánál és a kapott görbéket összeadni.

Az 56. egyenlet az időszetek hosszának eloszlását figyelembe véve tovább alakítható: a *scanek* (s) szerinti összegzés az $s \rightarrow \infty$ határesetben a spinhalmazok összes lehetséges életútját magában foglalja, méghozzá mindet olyan súllyal, amennyi az adott életút előfordulásának valószínűsége. A 37. egyenlet megadja annak a valószínűségét $(V_{\Delta t}^{(h)})$, hogy a spinhalmaz Δt ideig a $A^{(h)}$ konformációban van. Mivel az időszetek hossza egymástól független, egy életút csereidőpontjainak előfordulási valószínűsége az egyes időszetek előfordulási valószínűségének szorzata. Ezen felül figyelembe kell még venni az egyes időszetekben aktuális konformerek előfordulási valószínűségét is: az első időszetben K_{h_1} , a többiben $\kappa_{h_r, -h_r}$, ahol h_r annak a konformernek az indexe, amelyekben a spinhalmaz az r -edik időszetben van. Így adott konformersorrenddel a spektrum várható értéke egy adott frekvencián:

$$\dots \int_0^\infty \int_0^\infty \sum_{r=1}^\infty I^+ \frac{(1 - \exp((iL^{(h_r)} - k^*)\Delta t^{(r)})) K_{h_1} \prod_{r=2}^r \kappa_{h_r, -h_r} \prod_{r=1}^r V_{\Delta t^{(r)}}^{(h_r)})}{(k^* - iL^{(h_r)}) \exp((i\omega + k^*)t^{(r-1)})} \sigma^{(r-1)} d\Delta t^{(1)} d\Delta t^{(2)} \dots, \quad (57)$$

ezt a képletet kell még összegezni a konformerek előfordulásának összes variációjára. A spektrum véges ezért az 57. egyenletben az összes integrál véges, így az összegzés és az integrálok felcserélhetők. A $\Delta t^{(r)}$ szerinti integrálásban csak néhány tag szerepel (behelyettesítve $V_{\Delta t^{(r)}}^{(h_r)}$ értékét a 37. egyenlet alapján):

$$\int_0^\infty (1 - \exp((iL^{(r)} - k^*)\Delta t^{(r)})) \cdot d_{h_r} \exp(-d_{h_r} \Delta t^{(r)}) d\Delta t^{(r)} = \frac{k^* - iL^{(r)}}{d_{h_r} + k^* - iL^{(r)}}, \quad (58)$$

ahol az $L^{(r)}$ mátrixszal való osztást a nevező inverzével való szorzásként kell értelmezni (a tört számlálója és nevezője kommutál egymással). Így az r -edik időszetnek a spektrumhoz adott átlagos járuléka:

$$\bar{Y}^{(r)}(\omega) = I^+ \frac{K_{h_1} \prod_{r=2}^r \kappa_{h_r, -h_r}}{k^* + d_{j_r} - iL^{(h_r)}} \int_0^\infty \dots \int_0^\infty \left(\prod_{r=1}^{r-1} V_{\Delta t^{(r)}}^{(h_r)} \right) \frac{\sigma^{(r-1)}}{\exp((i\omega + k^*)t^{(r-1)})} d\Delta t^{(1)} \dots d\Delta t^{(r-1)}. \quad (59)$$

Az integrálon belüli rész az egyes időszetek hosszától függő tényezők szorzatára bomlik:

$$\left(\prod_{r=1}^{r-1} V_{\Delta t^{(r)}}^{(h_r)} \right) \frac{\sigma^{(r-1)}}{\exp((i\omega + k^*)t^{(r-1)})} = \left(\prod_{r=1}^{r-1} \left[V_{\Delta t^{(r)}}^{(h_r)} \exp((iL^{(h_r)} - k^*)\Delta t^{(r)}) \right] \right) \sigma^{(0)}. \quad (60)$$

Ezek a tényezők egymástól függetlenül integrálhatók, csak a szorzásuk sorrendje nem cserélhető fel. A $\Delta t^{(r)}$ szerinti integrálásban az 60. egyenletben szögletes zárójelben szereplő rész érintett:

$$\int_0^\infty \exp\left((iL^{\langle h_r \rangle} - k^*)\Delta t^{(r')}\right) d_{h_r} \exp\left(-d_{h_r} \Delta t^{(r')}\right) d\Delta t^{(r')} = \frac{d_{h_r}}{d_{h_r} + k^* - iL^{\langle h_r \rangle}}. \quad (61)$$

Így az átlagos járulékok:

$$\bar{Y}^{(r)}(\omega) = \frac{I^+}{d_{h_r}} \left(\prod_{r'=r}^2 \frac{\kappa_{j_{r'-1}j_r} d_{h_r}}{k^* + d_{h_r} - iL^{\langle h_r \rangle}} \right) \frac{K_{h_r} d_{h_r}}{k^* + d_{h_r} - iL^{\langle h_r \rangle}} \sigma^{(0)}. \quad (62)$$

Vezessük be a Ξ_h jelölést a következőképpen (ez összesen $m \times si$ darab mátrixot jelent):

$$\Xi_h = \frac{1}{k^* + d_h - iL^{\langle h \rangle}} = C^{(h)} \frac{k^* + d_h + i(\Omega^{(h)} - \omega)}{(k^* + d_h)^2 + (\Omega^{(h)} - \omega)^2} (C^{(h)})^\top. \quad (63)$$

A 62. egyenletbe a 39. egyenlet alapján behelyettesítve κ_{hg} definícióját a bomlási együtthatók (d_h) kiesnek és csak a sebességi együtthatók maradnak meg:

$$\bar{Y}^{(r)}(\omega) = I^+ \left(\prod_{r'=r}^2 k_{h_{r'-1}h_r} \Xi_{h_r} \right) K_{h_r} \Xi_{h_r} \sigma^{(0)}, \quad (64)$$

a teljes spektrumot pedig úgy kapjuk, hogy a 64. egyenletet összegezzük minden időszelet-re a konformerek összes lehetséges variációjában:

$$\bar{Y}(\omega) = I^+ \sum_{r=1}^\infty \sum_{h_1, h_2, \dots, h_r=1}^m \left(\prod_{r'=r}^2 k_{h_{r'-1}h_r} \Xi_{h_r} \right) K_{h_r} \Xi_{h_r} \sigma^{(0)}. \quad (65)$$

A 65. egyenlet alapján a spektrum elvileg minden egyes frekvenciára külön számolható. A képlet a gyakorlatban időigénye miatt azonban nehézkesen alkalmazható.

4.4. Lehetséges egyszerűsítések

4.4.1. Szimuláció a Hilbert-térben

Az 3.3.2. fejezetben a Liouville – von Neumann egyenletet (18. egyenlet) úgy oldottuk meg, hogy a Liouville-térbe transzformáltuk (így kaptuk az $L^{(r)}$ operátort) és az így keletkező differenciálegyenletrendszer oldottuk meg. A számolás azonban a Hilbert-térben is elvégezhető. A levezetéshez a 18. egyenlet egy másik, az irodalomból ismert megoldását használjuk [139]:

$$\rho(t + \tau) = \exp(-iH^{(r)}\tau) \rho(t) \exp(iH^{(r)}\tau). \quad (66)$$

Ez a megoldás akkor helyes, ha a Hamilton-operátor $(H^{(r)})$ állandó a $(t + \tau)$ és t időpont között, azaz jelen esetben egy időszeleten belül.

A 66. egyenletben az $\exp(\pm iH^{(r)}\tau)$ kifejezést kell részletezni. A Hamilton-operátor bázisfüggvényeinek $(\Psi_k^{(r)})$ bázisán mind az eredeti operátor, mind a keresett exponenciális operátor diagonális, azaz:

$$\exp(\pm iH^{(r)}\tau) = U^{(r)} \exp(\pm iA^{(r)}\tau) (U^{(r)})^T. \quad (67)$$

ahol $A^{(r)}$ jelöli a Hamilton-operátor sajátértékekből álló diagonális mátrixot, $U^{(r)}$ a 15. egyenletben definiált $u_{ai}^{(r)}$ együtthatókból felépített unitér mátrix, amelynek inverze a transzponáltja, $(U^{(r)})^T$. A 67. egyenlet jobb oldalát behelyettesítve a 66. egyenletbe:

$$\rho(t) = U^{(r)} \exp(-iA^{(r)}\Delta t) (U^{(r)})^T \rho^{(r-1)} U^{(r)} \exp(iA^{(r)}\Delta t) (U^{(r)})^T, \quad (68)$$

amely összefüggés alapján a $\rho(t)$ sűrűségmátrix elemei kiszámolhatók $\rho^{(r)}$ ismeretében. Így a teljes időszeleten át propagáláshoz a következő képletet használhatjuk:

$$\rho^{(r)} = U^{(r)} \exp(-iA^{(r)}\Delta t) (U^{(r)})^T \rho^{(r-1)} U^{(r)} \exp(iA^{(r)}\Delta t) (U^{(r)})^T. \quad (69)$$

A Hilbert-térben a detektált jelet a sűrűségmátrixból a következő képlettel számolhatjuk:

$$fid(t) = Tr(\rho(t)I^+). \quad (70)$$

A 70. egyenletbe behelyettesítve a 68. egyenletben kapott eredményt, és átrendezve:

$$fid^{(r)}(t) = Tr\left[\left((U^{(r)})^T I^+ U^{(r)}\right) \cdot \exp(-iA^{(r)}\Delta t) (U^{(r)})^T \rho^{(r)} U^{(r)} \exp(iA^{(r)}\Delta t)\right]. \quad (71)$$

A szögletes zárójelben lévő rész az I^+ operátor transzformált mátrixa $(A^{(r)})$:

$$A^{(r)} = U^{(r)} I^+ (U^{(r)})^T. \quad (72)$$

Az $A^{(r)}$ mátrix (j, k) elemének $(a_{jk}^{(r)})$ fizikai jelentése a $\psi_k^{(r)} \rightarrow \psi_j^{(r)}$ átmenet amplitúdója.

A 72. egyenletet behelyettesítve a 71.-be:

$$fid^{(r)}(t) = Tr\left(A^{(r)} \exp(-iA^{(r)}\Delta t) (U^{(r)})^T \rho^{(r-1)} U^{(r)} \exp(iA^{(r)}\Delta t)\right). \quad (73)$$

A teljes $fid(F)$ az időszelletek fid jeinek uniójaként kapott egyedi scanek összege, hasonlóan a 51–52. egyenletekhez.

A sűrűségmátrix Liouville- és Hilbert-térbeli kifejezése ekvivalens egymással (ld. 4.4.4. fejezet). A kettő közötti jelentős különbséget a propagáló mátrixok mérete jelenti.

4.4.2. Transzformált sűrűségmátrix

Vezessük be a $\xi^{(r)}(t)$ jelölést a transzformált sűrűségmátrixra a következőképpen:

$$\xi^{(r)}(t) = (U^{(r)})^T \rho(t) U^{(r)}. \quad (74)$$

Ekkor a $\xi^{(r)}(t)$ a sűrűségmátrix a Hamilton-operátor sajátfüggvényeinek bázisán. Az 68. egyenletbe behelyettesítve a 74. egyenlet alapján és átrendezve:

$$\xi^{(r)}(t) = \exp(-iA^{(r)}\Delta t) \xi^{(r)}(t^{(r-1)}) \exp(iA^{(r)}\Delta t). \quad (75)$$

Egy teljes időszeleten át propagáltatva:

$$\xi^{(r)}(t^{(r)}) = \exp(-iA^{(r)}\Delta t^{(r)}) \xi^{(r)}(t^{(r-1)}) \exp(iA^{(r)}\Delta t^{(r)}), \quad (76)$$

majd áttérve az új $(r+1)$ -edik időszelet sajátfüggvényeinek bázisára:

$$\xi^{(r+1)}(t^{(r)}) = (U^{(r+1)})^T \left[U^{(r)} \xi^{(r)}(t^{(r)}) (U^{(r)})^T \right] U^{(r+1)}. \quad (77)$$

A további propagálást ebből a mátrixból lehet számolni. A detektált jelet a sűrűségmátrixból a következő képlettel számolhatjuk:

$$fid^{(r)}(t) = Tr(A^{(r)} \exp(-iA^{(r)}\Delta t) \xi^{(r)}(t^{(r-1)}) \exp(iA^{(r)}\Delta t)), \quad (78)$$

ahol $t^{(r-1)}$ és $t^{(r)}$ között van.

A mátrixok exponenciális függvényeinek kiszámításának leggyakoribb és általában leggyorsabb módja az átnormálást követő Pade-approximáció, majd ezen részeredmény hatványozása [140, 141]. Ez a megoldás egy-egy mátrix esetében lényegesen gyorsabb, mint a mátrix diagonalizálása és ezt követően a diagonális alak exponenciális függvényének kiszámolása, azaz az a módszer, amit jelen esetben használunk. Jelen szimuláció specialitása az, hogy csak néhány mátrixot kell diagonalizálni viszont sokszor kell számolni különböző mátrixok exponenciálisát (amely mátrixok általában egymás nem egész számú többszörösei). Ilyen feltételek mellett a feladat kevesebb művelettel megoldható a 75–78. egyenletek használatával, mint az exponenciálisok Pade-approximációjával.

4.4.3. Blokkdiagonalizáció

A 3.3.1.2. fejezetben láttuk, hogy a 41. egyenletben definiált $H^{(r)}$ mátrix $n+1$ kisebb négyzetes blokk direkt összege (n a spinhalmazban lévő magok száma. Emiatt az $U^{(r)}$ valamint az $A^{(r)}$, $\xi^{(r)}$ mátrixok is blokkosíthatók. Egy n magból álló spinhalmaz esetén a $\xi^{(r)}$ sűrűségmátrix n darab blokkra bomlik, amelyek mérete adott $l = 1 \dots n$ index esetén $\begin{pmatrix} n \\ l-1 \end{pmatrix} \times \begin{pmatrix} n \\ l \end{pmatrix}$. Hasonlóképpen bomlanak fel az $A^{(r)}$ mátrixok is. A 76–78. egyenletben sze-

replő $U^{(r)}$ és $A^{(r)}$ mátrixok $\xi^{(r)}$ előtt és után a teljes mátrix különböző l indexhez tartozó blokkjait jelentik, így a ezek az egyenletek a következő alakot öltik:

$$\xi_l^{(r)}(t^{(r)}) = \exp(-iA_{l-1}^{(r)}\Delta t)\xi_l^{(r)}(t^{(r-1)})\exp(iA_l^{(r)}\Delta t), \quad (79)$$

$$\rho_l^{(r)} = U_{l-1}^{(r+1)}\xi_l^{(r)}(t^{(r)})\left(U_l^{(r)}\right)^T \text{ és } \xi_l^{(r+1)}(t^{(r)}) = \left(U_{l-1}^{(r+1)}\right)^T \rho_l^{(r)}U_l^{(r+1)}, \quad (80)$$

illetve

$$fid_l^{(r)}(t) = \text{Tr}\left(A_l^{(r)}\exp(-iA_{l-1}^{(r)}\Delta t)\xi_l^{(r)}(t^{(r-1)})\exp(iA_l^{(r)}\Delta t)\right). \quad (81)$$

A szimuláció szempontjából a blokkosítás előnye, hogy a szimuláció minden egyes blokkra külön végezhető, ezért a 2^n méretű sűrűségmátrixok helyett kisebb blokkokkal lehet számolni.

4.4.4. A Hilbert- és Liouville térbeli leírás ekvivalenciája

A 4.3. és a 4.4.1. fejezetben egy spinhalmaz által generált fidet két különböző módon vezettük le. Ebben a fejezetben a két leírásból származó képletek ekvivalenciáját bizonyítjuk.

4.4.4.1. A bázis- és sajátfüggvények kapcsolata a Hilbert-térben

A $\rho(t)$ sűrűségmátrix elemeit $\rho_{ab}(t)$ jelöli a φ_a szorzatfüggvények bázisán. A 15. egyenlethez hasonlóan az egyes időszeletek Hamilton-operátorának sajátfüggvényei ($\psi_k^{(r)}$) felírhatók a szorzatfüggvények lineáris kombinációjaként:

$$\psi_k^{(r)} = \sum_a u_{ak}^{(r)} \varphi_a, \quad (82)$$

azaz az $u_{ak}^{(r)}$ együtthatók definíciója:

$$u_{ak}^{(r)} = \langle \varphi_a | \psi_k^{(r)} \rangle = \langle \psi_k^{(r)} | \varphi_a \rangle. \quad (83)$$

A $\psi_k^{(r)}$ sajátfüggvények is ortonormált bázisát alkotják a Hilbert-térnek, így a szorzatfüggvények felírhatók ezek lineáris kombinációjaként:

$$\varphi_a = \sum_k u_{ak}^{(r)} \psi_k^{(r)}, \quad (84)$$

ahol az $u_{ak}^{(r)}$ együtthatók megegyeznek a 82. egyenletben szereplő együtthatókkal, mivel az $u_{ak}^{(r)}$ valós együtthatókból álló $U^{(r)}$ mátrix unitér.

Az r -edik időszület Hamilton-operátorának k -adik sajátértékeit $\lambda_k^{(r)}$ jelöli:

$$H^{(r)}\psi_k^{(r)} = \lambda_k^{(r)}\psi_k^{(r)}. \quad (85)$$

A $\psi_k^{(r)}$ és a $\psi_j^{(r)}$ közötti átmenet frekvenciája ekkor $\lambda_j^{(r)} - \lambda_k^{(r)}$, amplitúdója pedig:

$$a_{kj}^{(r)} = \sum_{a,b} u_{ak} u_{bj} I_{ab}. \quad (86)$$

A detektált jel intenzitása $(Y_{ij}^{(r)})$ az $a_{ij}^{(r)}$ amplitúdó négyzete lenne (vö. 17. egyenlet).

4.4.4.2. A bázis- és sajátfüggvények kapcsolata Liouville-térben

A $\sigma(t)$ vektor a $\sigma_e(t)$ együtthatókkal írható fel a Φ_e bázisfüggvények lineáris kombinációjaként:

$$\sigma(t) = \sum_e \sigma_e(t) \Phi_e, \quad (87)$$

azaz a $\sigma_e(t)$ együtthatókat a következő összefüggés definiálja:

$$\sigma_e(t) = \langle \Phi_e | \sigma(t) \rangle, \quad (88)$$

ahol Φ_e jelöli az átmenetek terében az e -edik báziskoherencziát, (e a bázisfüggvények indexe, értelmezési tartománya 1 és n^2 között van).

Az $L^{(r)}$ mátrix sajátfüggvényeit $\Psi_p^{(r)}$ -vel (p a sajátfüggvények indexe, 1 és M között változik), a $\Psi_p^{(r)}$ -hez tartozó sajátértéket $\omega_p^{(r)}$ -vel jelölve:

$$L^{(r)} \Psi_p^{(r)} = \omega_p^{(r)} \Psi_p^{(r)}, \quad (89)$$

ahol a felső indexben lévő r az aktuális időszelet indexe (ezek a mennyiségek a konformertől és emiatt az időszelettől függenek). Az $\mathcal{L}^{(r)}$ diagonális mátrix (44. egyenlet) ezen $\omega_p^{(r)}$ értékeket tartalmazza. A $C^{(r)}$ mátrix (44. egyenlet) elemei $(c_{ep}^{(r)})$ a $L^{(r)}$ operátor sajátfüggvényeinek $(\Psi_p^{(r)})$ lineárkombinációs együtthatói a szorzatfüggvények bázisán:

$$\Psi_p^{(r)} = \sum_e c_{ep}^{(r)} \Phi_e, \quad (90)$$

azaz a $c_{ep}^{(r)}$ együtthatókat a következő egyenlet definiálja:

$$c_{ep}^{(r)} = \langle \Phi_e | \Psi_p^{(r)} \rangle = \langle \Psi_p^{(r)} | \Phi_e \rangle. \quad (91)$$

Hasonlóképpen a Φ_e bázisátmenetek is felírhatók a $\Psi_p^{(r)}$ sajátfüggvények bázisán:

$$\Phi_e = \sum_p c_{ep}^{(r)} \Psi_p^{(r)}, \quad (92)$$

ahol a $c_{ep}^{(r)}$ együtthatók ugyanazok, mint a 90. egyenletben.

A 50. egyenlettel definiált $Q^{(r)}$ mátrix p -edik eleme $(o_p^{(r)})$ fizikailag a p -edik saját-függvény által adott jel amplitúdója:

$$o_p^{(r)} = \langle \hat{I}^+ | \Psi_p^{(r)} \rangle = \sum_p c_{ep}^{(r)} \phi_e, \quad (93)$$

ahol $\phi_e = 1$, ha a Φ_e átmenet megengedett és nulla egyébként.

4.4.4.3. Az ekvivalencia bizonyítása az együttthatók segítségével

A bizonyításhoz először a Hilbert-tér $u_{ak}^{(r)}$ és a Liouville-tér $c_{ep}^{(r)}$ együttthatói, valamint a $H^{(r)}$ és az $L^{(r)}$ mátrixok sajátértékei közötti kapcsolatot kell megteremteni. Belátható, hogy ha az e ill. p indexek Hilbert-térbeli megfelelői az (a, b) és (k, j) indexpárok, akkor az együttthatók között a

$$c_{ep}^{(r)} = u_{ak}^{(r)} u_{bj}^{(r)} \quad (94)$$

kifejezés teremt kapcsolatot, a sajátértékek között pedig az

$$\omega_p^{(r)} = \lambda_j^{(r)} - \lambda_k^{(r)} \quad (95)$$

összefüggés áll fenn. Ez utóbbit alátámasztja az is, hogy az átmenetek frekvenciája nem függhet attól, hogy melyik térben határozzuk meg. Hasonlóképpen teljesül az amplitúdók egyenlősége is:

$$q_p^{(r)} = a_{kj}^{(r)}. \quad (96)$$

A Liouville-térben a precessziót leíró 46. egyenlet mátrixszorzásait a 91. egyenletben definiált változókkal helyettesítve a sűrűségmátrix f -edik eleme $(\sigma_f(t))$ a következő képpel írható fel:

$$\sigma_f(t) = \sum_{p,e} c_{fp}^{(r)} c_{ep}^{(r)} \exp(i\omega_p^{(r)} \Delta t) \sigma_e^{(r-1)}. \quad (97)$$

Az ennek megfeleltethető a 68. egyenletbe a 83. egyenletben definiált koefficienseket behelyettesítve a következő kifejezést kapjuk a sűrűségmátrix egyes $\rho_{ab}(t)$ elemeire:

$$\rho_{ab}(t) = \sum_{c,d,k,j} u_{ak}^{(r)} u_{bj}^{(r)} \cdot u_{ck}^{(r)} u_{dj}^{(r)} \cdot \exp(i(\lambda_j^{(r)} - \lambda_k^{(r)}) \Delta t) \cdot \rho_{cd}^{(r)}. \quad (98)$$

Az 97. és a 98. egyenlet ekvivalens, ami könnyen belátható a 19., a 94. és a 95. egyenletek alapján, ha figyelembe vesszük, hogy az e, f , ill. p Liouville-térbeli indexek megfelelői a (a, b) , (c, d) és (k, j) Hilbert-térbeli indexpárok.

Az időszeleten át történő propagálás képletei (47. és 69. egyenlet) az általános propagálástól csak a Δt változó $\Delta t^{(r)}$ -re való cseréjében különbözik, így a 46. és a 68. képlet ekvivalenciájából következik, hogy a két egyenlet tartalma megegyezik.

A Liouville-térben a detektálásra kapott 50. egyenletben a mátrixszorzásokat a 91. egyenlet alapján koeficiensekre cserélve:

$$fid^{(r)}(t) = \sum_{p,e} o_p^{(r)} c_{ep}^{(r)} \exp(i\omega_p^{(r)} \Delta t) \rho_e^{(r-1)}. \quad (99)$$

Az ennek megfelelő képlet Hilbert-térben a 73. egyenlet. Ennek a képletnek a mátrixszorzásait 83. egyenletben definiált koeficiensekkel felírva és a szorzótényezőket átrendezve:

$$fid(t) = \sum_{a,b,k,j} a_{kj}^{(r)} \cdot u_{ak}^{(r)} u_{bj}^{(r)} \cdot \exp(i(\lambda_j^{(r)} - \lambda_k^{(r)}) \Delta t) \cdot \rho_{ab}^{(r)}. \quad (100)$$

A 99. és a 100. egyenletek ekvivalenciája a 19., a 94., a 95. és a 96. összefüggésekből egyértelmű.

4.5. Az átlag- és az egyedi sűrűségmátrix szimulációjának összevetése

4.5.1. Mátrixok mérete

A szimulációs programok memóriaigényét mindkét módszernél alapvetően a propagáló mátrixok (az $i\bar{L} + \bar{R} + \bar{X}$ mátrix illetve a $H^{(r)}$ mátrixok) mérete határozza meg.

A szimulációkban a detektált jel csak az egykvantumkoherenciáktól függ, így a sűrűségmátrixnak csak ezekkel az elemeivel kell foglalkozni, ami a szimulációs programok memóriaigényét csökkenti: a mátrixméret a Liouville-tér 2^{2N} dimenziója helyett csak

$$M = \begin{pmatrix} 2N \\ N-1 \end{pmatrix}. \quad (101)$$

A Hamilton-operátor totálspinkvantszám szerinti blokkosítása tovább mérsékli a mátrixok dimenzióját:

$$\max(M_t) = \begin{pmatrix} N \\ N/2 \end{pmatrix}, \quad (102)$$

ahol N a szimulációban figyelembe vett összes spin száma, azaz általában $N = n \cdot m$, n a spinhalmaz mérete és m a konformerek száma.

Az átlag-sűrűségmátrixon alapuló szimulációkban további méretcsökkentés érhető el az intermolekuláris (azaz konformerek közötti) átmenetek elhanyagolásával. Ezáltal a mátrixok dimenziója

$$\max(M_I) = m \cdot \binom{n}{n/2} \cdot \binom{n}{n/2+1} \quad (103)$$

méretre csökken, ami azonban még mindig $m \cdot 4^n$ nagyságrendű. Nem kölcsönös csere esetén ez az átlag-sűrűségmátrix kiszámításán alapuló szimulációk mátrixainak dimenziója. Kölcsönös csere esetén bizonyos spinek egybeesnek, ami a mátrixméretet tovább csökkenti, legjobb esetben a 103. egyenletben $m = 1$ lesz.

A Monte Carlo szimulációban az intermolekuláris átmeneteket eleve elhagyjuk, mivel a mátrixban csak egyetlen spinhalmaz átmenetei vannak jelen, amelyek egyszerre csak egy konformerben fordulhatnak elő. Ezen felül a szimuláció a Liouville-tér helyett a kisebb dimenziójú Hilbert-térben is elvégezhető (ld. 4.4 fejezet), így a propagáló mátrix maximális mérete

$$n' = \binom{n}{n/2}, \quad (104)$$

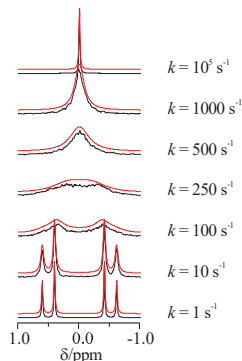
ami lényeges csökkenés mind a teljes mérethez, mind az átlag-sűrűségmátrixon alapuló szimulációkhoz képest.

A gyakorlati szimulációkban ezt jelenti, hogy míg az átlag-sűrűségmátrixok szimulációjában a spinrendszer méretének felső korlátja 15 mag körül van (körülbelül hétmagos spinhalmaz kettő, vagy ötmagos három konformerrel), addig a Monte Carlo módszerrel ennél lényegesen nagyobb rendszerek szimulálhatók (pl. 6.1. fejezet).

4.5.2. Egy kétmagos kölcsönös cserefolyamat

Az átlag- és az egyedi sűrűségmátrixos leírást egy kétmagos kölcsönös cserefolyamatban részt vevő spinrendszeren hasonlítjuk össze. A vizsgált rendszerben a két cserélő mag egymással csatolásban van. Ebben az esetben a spinrendszer és spinhalmaz is két magból áll (ld. 5. ábra).

A 6. ábrán egy konkrét AB csatolt rendszer szimulált spektrumai láthatók néhány cseresebességnél az átlag-sűrűségmátrix illetve a Monte Carlo módszerrel szimulálva (közös adatok: $\Delta\nu = 1,0$ ppm, azaz 100 Hz, $J = 10$ Hz). Látható, hogy a spektrumok minden cseresebességnél



6. Ábra Kölcsönösen cserélő AB spinrendszer spektrumai néhány hőmérsékleten a ProMoCS (piros) ill. MEXICO (fekete) programmal szimulálva.

azonosak. A kétféle megközelítéssel a jelalakváltozás egyes aspektusai különböző szempontból magyarázhatók.

4.5.2.1. Az AB kölcsönös csere leírása átlag sűrűségmátrix segítségével

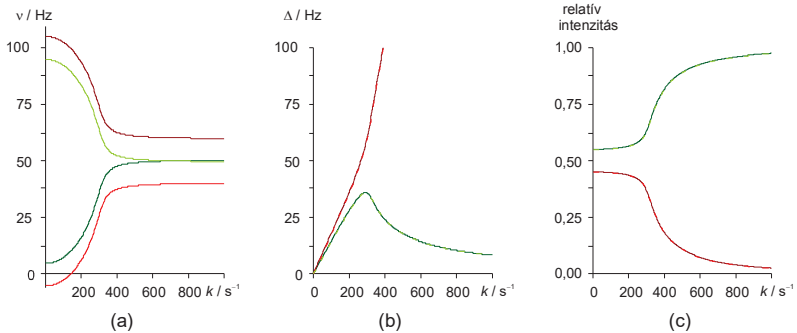
Az átlag sűrűségmátrix szimulációjához a 28. egyenletben zárójelben szereplő mátrixot kell felépíteni és annak a sajátértékeit meghatározni. A relaxációt leíró \bar{R} mátrixot eddig is elhanyagoltuk, a maradék két mátrix összege a 20. és a 24. egyenletek alapján:

$$(\bar{L} + \bar{X})_{1,2} = \begin{pmatrix} \pi(2\nu_2 \pm J) - k & \mp i\pi J + k \\ \mp i\pi J + k & \pi(2\nu_1 \pm J) - k \end{pmatrix}, \quad (105)$$

ahol az $\bar{L} + \bar{X}$ mátrix két indexe a két szimulált blokkot jelenti, a \pm jelek közül a felső az 1, az alsó a 2 indexű blokkhoz tartozik. A két mátrixblokk sajátértékei analitikusan meghatározhatók:

$$\bar{\omega}_{1,2} = \pi(\nu_1 + \nu_2 \pm J) - k \pm i\sqrt{(\pi(\nu_1 - \nu_2))^2 - (k \mp \pi J)^2}, \quad (106)$$

ahol a gyökjel előtti \pm az azonos blokkhoz tartozó két sajátértéket különbözteti meg, a másik kettő a 105. egyenletből öröklődött. Ezen sajátértékek valós és képzetes részének változását, azaz a jelek frekvenciáját és szélességét mutatja a 7. ábra. Látható, hogy a négyből kettő (a két blokkból egy-egy) sajátérték valós része monoton nő, azaz ezek a jelek eltűnnek az alapvonalban. A maradék két sajátérték valós részének maximuma van (ez a koaleszcencia), ez után a pont után egyre élesebbek lesznek. E két sajátérték képzetes része a koaleszcencia után a cseresebesség növelésével gyakorlatilag azonosává válik. Ezzel szemben a két kiszélesedő jel közepe más frekvenciára áll be, de kiszélesedésük mellett az



7. Ábra A szimulált kétmagos kölcsönös cserefolyamat spektrumában kapott jelek jellemző adatainak változása a cseresebesség függvényében (a) frekvencia (b) félérték szélesség (c) relatív intenzitás. A sötét vonalak a 0-1, a világosak az -1-0 totális pínkvantumszámú altérben számolt adatokat mutatják.

intenzitásuk is nullára csökken (7.c ábra). Ezek alapján tehát az AB spinrendszer négy jele közül kettő eltűnik és csak a maradék kettő egyesül.

A fentiekből következik, hogy a kölcsönösen cserélő AX spinrendszer sem modellezhető egyszerűen a hagyományos vektormodelles képben. A sűrűségmátrixos leírásban a csatolás erőssége nem számít, így gyenge csatolás esetén is a 7. ábrán bemutatotthoz hasonló görbékert kapnánk. A vektormodelles közelítésben a két altér frekvenciái váltakoznak, aminek eredménye a gyors csere határesetében a két sötétebb és a két világosabb vonalpár átlaga lenne, amelyek nem esnek egybe. Így ez a modell a szingulett helyett dubletet eredményezne a gyors csere esetében, ami nyilvánvalóan hibás.

4.5.2.2. A gyors csere leírása egyedi sűrűségmátrixszal

A kétmagos kölcsönös cserefolyamat esetén a szokásos jelölésekkel az első konformer L operátorának (ld. 43. egyenlet) két blokkja a következő alakban írható fel (az áttekinthetőség kedvéért a blokkok indexét elhagytuk, ezekre a mátrixelemekben szereplő \pm utal):

$$L^{(1)} = 2\pi \begin{pmatrix} \nu_1 \pm \frac{J}{2} & \mp \frac{J}{2} \\ \mp \frac{J}{2} & \nu_2 \pm \frac{J}{2} \end{pmatrix}, \quad (107)$$

ahol a frekvenciáknál és a csatolási állandónál a konformer indexét az áttekinthetőség érdekében elhagytuk. A második konformer mátrixa hasonló, csak a két mag szerepét meg kell cserélni (azaz a szokásos jelölésekkel $\nu_1^{(2)} = \nu_2$ és $\nu_2^{(2)} = \nu_1$):

$$L^{(2)} = 2\pi \begin{pmatrix} \nu_2 \pm \frac{J}{2} & \mp \frac{J}{2} \\ \mp \frac{J}{2} & \nu_1 \pm \frac{J}{2} \end{pmatrix}. \quad (108)$$

A t időpontban detektált jelet a 49. egyenlet alapján lehet kiszámolni, ahol t az r -edik időszletben van (azaz $t^{(r-1)} < t < t^{(r)}$). Ez az egyenlet röviden a következő formába írható (49. egyenlet a rövidebb alakját használva):

$$fid(t) = I^+ \exp(iL^{(r)}\Delta t) \sigma^{(r-1)}. \quad (109)$$

A 109. egyenletben a $\sigma^{(r-1)}$ helyébe beírva a rekurzív definícióját (47. egyenlet) kapjuk a teljes kifejezést a detektált jelle:

$$fid(t) = I^+ \exp(iL^{(r)}\Delta t) \exp(iL^{(r-1)}\Delta t^{(r-1)}) \dots \exp(iL^{(1)}\Delta t^{(1)}) \sigma^{(0)}. \quad (110)$$

A 110. egyenletben az $L^{(1)}, L^{(2)}, \dots, L^{(r)}$ mátrixok a két konformer $L^{(1)}$ ill. $L^{(2)}$ operátorai közül kerülnek ki. Feltételezve, hogy az első időszletben a szimulált spinhalmaz az

$A^{(1)}$ konformerben van és r páros, az időszeletek helyett a konformerek mátrixait behelyettesítve a 110. egyenlet a következő alakot ölti:

$$fid(t) = I^+ \exp(iL^{(1)}\Delta t) \exp(iL^{(2)}\Delta t^{(r-1)}) \dots \exp(iL^{(1)}\Delta t^{(1)}) \sigma^{(0)}. \quad (111)$$

$L^{(1)}$ és $L^{(2)}$ nem kommutáló mátrixok, ezért

$$\exp(iL^{(1)}\Delta t^{(1)}) \exp(iL^{(2)}\Delta t^{(2)}) \neq \exp(iL^{(1)}\Delta t^{(1)} + iL^{(2)}\Delta t^{(2)}), \quad (112)$$

de a két oldal különbsége kis $\Delta t^{(r)}$ értékre elhanyagolható [16]. Gyors cserefolyamat esetén a $\Delta t^{(r)}$ értékek minden r értékre kicsik, így a 111. egyenlet bal oldalát jól közelíti a következő kifejezés:

$$fid(t) = I^+ \exp(iL^{(1)}\Delta t + iL^{(2)}\Delta t^{(r-1)} + \dots + iL^{(1)}\Delta t^{(1)}) \sigma^{(0)}. \quad (113)$$

A vizsgált cserefolyamat kölcsönös, ezért az $A^{(1)}$ konformerben töltött összes idő várható értéke megegyezik az $A^{(2)}$ -ben töltött idővel, így az 113. egyenletben az összegzés egyszerűbb alakra hozható (mivel a mátrixok összeadása kommutatív):

$$fid(t) = I^+ \exp\left(i \frac{L^{(1)} + L^{(2)}}{2} t\right) \sigma^{(0)}. \quad (114)$$

Az 114. egyenletben az exponenciális argumentumába a 107. és a 108. egyenletben definiált operátorok átlaga kerül, ami a következő:

$$\frac{L^{(1)} + L^{(2)}}{2} = \pi \begin{pmatrix} \nu_1 + \nu_2 \pm J & \mp J \\ \mp J & \nu_2 + \nu_1 \pm J \end{pmatrix}. \quad (115)$$

Ez az operátor formálisan megegyezik egy egyszerű A_2 spinrendszer Hamilton operátorával (ahol az A mag frekvenciája $\frac{1}{2}(\nu_1 + \nu_2)$ és csatolási állandó J). Ennek a mátrixnak a sajátértékei $\frac{1}{2}(\nu_1 + \nu_2)$ (degenerált) és $\frac{1}{2}(\nu_1 + \nu_2) \pm J$, az utóbbiak intenzitása a spektrumban nulla. Magas hőmérsékleten, ahol gyors a csere, ez a mátrix egyetlen szingulettet eredményez, ahogy az a kísérleti adatok alapján elvárható.

4.5.2.3. A direkt képlet értelmezése kölcsönös csere esetén

A kölcsönösen cserélő csatolt AB spinrendszer esetében a spektrum számolására kapott 65. egyenlet több ponton egyszerűsíthető. Egyrészt, a kölcsönös csere miatt csak egyetlen sebességi együttható van és az egymást követő cserékben a termék egyértelmű, így a konformerek szerinti összegzések közül csak az első időszelvény szerinti marad meg. Az egyensúlyi relatív koncentráció mindkét konformer esetében $\frac{1}{2}$, ez a tag kiemelhető az összegzésből. Így a maradék egyenlet:

$$\bar{Y}(\omega) = \frac{I^+}{2} \left\{ \sum_{r=1}^{\infty} \sum_{h_1=1}^2 \left(\prod_{r'=r}^2 k \mathcal{E}_{h_{r'}} \right) \mathcal{E}_{h_1} \right\} \sigma^{(0)}, \quad (116)$$

ahol a \mathcal{E}_{h_1} tényező a 107–108. egyenletekben szereplő mátrix alapján felépíthető.

Kis k értékeknél a \mathcal{E}_{h_1} mátrix egyes elemeinek valós és képzetes része ω függvényében keskeny abszorpciós, illetve diszperziós Lorentz-görbék összege. Két vagy több ilyen mátrix szorzata a görbék kis átfedése miatt gyorsan eltűnik, így ebben az esetben a 116. egyenletben kapcsos zárójelben lévő mátrixban az a tag az uralkodó, amelyikben a 116. egyenlet zárójeles része 1. Így a spektrumban a két éles jel jelenik meg.

A sebességi együttható növelésével a \mathcal{E}_{h_1} mátrix Lorentz-görbéi szélesednek. Ennek következtében a kétféle frekvenciához tartozó jel átfedése nő, a szorzatok lassabban tűnik el. A valós tagok összegzésében a görbék képzetes részeinek szorzata is megjelenik, és ezen görbék maximuma közeledik egymáshoz, így a szorzat egy maximumos görbe lesz. Az r szerinti összegzésben az első időszelvény szerepe lecsökken, egyre több tag szerepe érvényesül az összegzésben (a végtelen összegzésnek nyilvánvaló határt szab a relaxációt helyettesítő exponenciális tag). A magasabb sorszámú időszelvények spektrumaiban a görbék képzetes részeinek szorzataiból származó egyetlen csúcspot tartalmazó tagok egyre nagyobb szerephez jutnak. Ezek a csúcsok annál keskenyebbek, minél nagyobb bennük a képzetes tényezők száma. Így a jel összeolvadása, majd kiélesedése ezeknek a képzetes görbékől származó tagoknak köszönhető, a jelek valós részeinek szorzata eltűnik.

4.6. A vektormodell és a dinamikus folyamatok

4.6.1. Csatolt spinrendszerek a vektormodellben

Ebben a fejezetben a vektormodell egy olyan kibővített értelmezéséről lesz szó, amely segítségével a dinamikus folyamatok csatolt spinrendszerekre is értelmezhetők. Ezen felül megteremtjük a jelenség egyedi sűrűségmátrixos leírás és a kibővített vektormodell között.

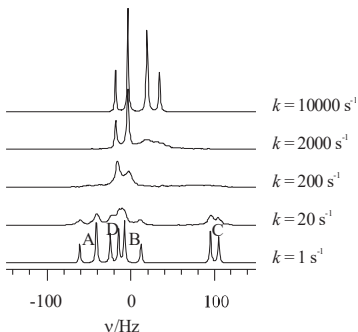
A gyengén csatolt spinrendszerekben a szorzatfüggvényeknek ($\varphi_1 = \alpha\alpha...\alpha$, $\varphi_2 = \alpha\alpha...\beta$, ..., $\varphi_{n'} = \beta\beta...\beta$, ahol $n' = 2^n$ a Hilbert-tér dimenziója) meghatározott energiájuk van, így az általuk definiált $\Phi_e = \varphi_a \rightarrow \varphi_b$ egykvantum-koherenciákhoz karakterisztikus precessziós frekvenciák rendelhetők. Ennek következtében ezek a Φ_e koherenciák a rendszer sajátállapotai, azaz a precesszió során populációjuk nem változik meg. A cserefo-

lyamatok nem változtatják meg a báziskoherenciákat, így gyengén csatolt esetben a Φ_e állapotok végig megmaradnak, és a karakterisztikus frekvenciájuk segítségével az általuk adott jel egyesével szimulálható. A vektormodellben a szorzatfüggvény-párok, azaz a báziskoherenciák mindegyikéhez rendelhető egy-egy kúp és gyengén csatolt rendszerekben ezek a kúpok egymástól függetlennek tekinthetők.

Erősen csatolt esetben a Φ_e koharenciáknak nincs karakterisztikus frekvenciájuk és emiatt a precesszió során egy spinhalmaz adott pillanatban az M -féle báziskoharencia bármelyikébe átkerülhet. A Φ_e báziskoharenciák helyett az egyes konformerek $\Psi_p^{(r)}$ sajátátmenetei azok, amik a precesszió során megmaradnak és karakterisztikus frekvenciájukkal precesszálnak. Ezek a $\Psi_p^{(r)}$ koharenciák attól függnnek, hogy a spinhalmaz melyik konformerében vannak (erre utal az időszeletet indexelő r a felső indexben), így a csere pillanatában szerepük nem értelmezhető. Ezen kettősség miatt a vektormodellben két „kúpsorozatot” kellene definiálni, egyet a bázis- és egyet a sajátfüggvények szerint. A továbbiakban a vektormodellbeli kúpokból csak a rajtuk lévő vektorok eredőjét vizsgáljuk.

Az egy magra definiált vektormodell eredő vektora (a makroszkopikus mágnesezettség vektor) egyben az egyetlen magból álló spinhalmaz sűrűségmátrixának vektoros reprezentációja is. Ez a sűrűségmátrix egyetlen komplex szám, ezért a vektormodellben is egyetlen vektor jelenik meg. Több mag esetén a sűrűségmátrix nagyobb, ennek megfelelően a vektormodellben egyetlen spinhalmazhoz több vektor tartozik. Ezen felül, egy többmagos spinrendszer esetén a sűrűségmátrix a megfelelő altér különböző bázisain (ezek a bázisok jelen esetben a Ψ és Φ függvényhalmazok) más alakot ölt, ezért a vektormodellben egy adott állapotra is többféle vektor jelenhet meg.

A vektormodell ezen kiterjesztése arra irányul, hogy kapcsolatot teremtsünk a precesszió és a csere során jellemző kétféle koharencia-típus között. E kapcsolat matematikai részleteiről a 4.3 fejezetben volt szó, ebben a részben az ott leírt matemati-



8. Ábra Egy kétmagos AB - CD cserét mutató csatolt spinrendszer szimulált spektrumai növekvő cseresebesség mellett. (1000 scan, 512 spektrumpont).

kai képleteket a vektormodell szintjén értelmezzük.

A szemléltetéshez egy általános nem kölcsönös $AB \leftrightarrow CD$ cserefolyamatot elemzünk.

A 8. ábra egy ilyen spinrendszer szimulált spektrumait mutatja. (A szimuláció paramétereit

a 1. táblázat tartalmazza). $k = 1 \text{ s}^{-1}$ esetén két átfedő dublett-dublett pár látható (az AB és CD rendszereknek megfelelően), melyek k növelésével, koaleszcenciát követően $k = 10000 \text{ s}^{-1}$ esetén egyetlen dublett-dublettet adnak.

A rendszerben jelenlévő két konformer az $A^{(1)}$ (az AB-vel jelölt) és az $A^{(2)}$ (CD-vel jelölt). A cserefolyamat statisztikai leírása a vektormodell szempontjából csak annyiban érdekes, hogy adott időpillanatokban a spinhalmaz konformert vált.

A két spinből álló spinhalmazhoz négy báziskoherencia tartozik ($\Phi_1 = \alpha\beta \rightarrow \alpha\alpha$, $\Phi_2 = \beta\alpha \rightarrow \alpha\alpha$, $\Phi_3 = \beta\beta \rightarrow \alpha\beta$ és $\Phi_4 = \beta\beta \rightarrow \beta\alpha$). Emellett minden konformernek négy sajátfüggvénye van ($\Psi_1^{(h)}$, $\Psi_2^{(h)}$, $\Psi_3^{(h)}$ és $\Psi_4^{(h)}$, ahol h a konformer indexe), amelyek karakterisztikus frekvenciáit $\omega_p^{(h)}$ -vel jelöltük ($p = 1 \dots 4$ a sajátátmenetek indexe). A blokkdiagonalizáció során (4.4.3. fejezet) a báziskoherenciák által kifeszített altér két két-dimenziós invariáns altérre bomlik, ebből a konkrét példa az első blokkot mutatja be (így a továbbiakban a p és e indexek értéke 1 vagy 2 lehet).

Jelölje $\sigma_e(t)$, illetve $\zeta_p^{(r)}(t)$ komplex szám a Φ_e bázisfüggvényhez, ill. $\Psi_p^{(r)}$ sajátfüggvényhez rendelt vektort a t időpontban (9. ábra). E két szám (vektor) közötti átváltás formailag a függvényekhez hasonló (90. ill. 92. egyenlet):

$$\zeta_p^{(r)}(t) = \sum_e c_{ep}^{(r)} \sigma_e(t) \text{ ill. } \sigma_e(t) = \sum_p c_{ep}^{(r)} \zeta_p^{(r)}(t). \quad (117)$$

Adott (r -edik) időszakban a két sajátfüggvényhez rendelt vektorok (azaz $\zeta_p^{(r)}$, ahol $p = 1, 2$) $\omega_p^{(r)}$ frekvenciával precesszál (9. ábra A része):

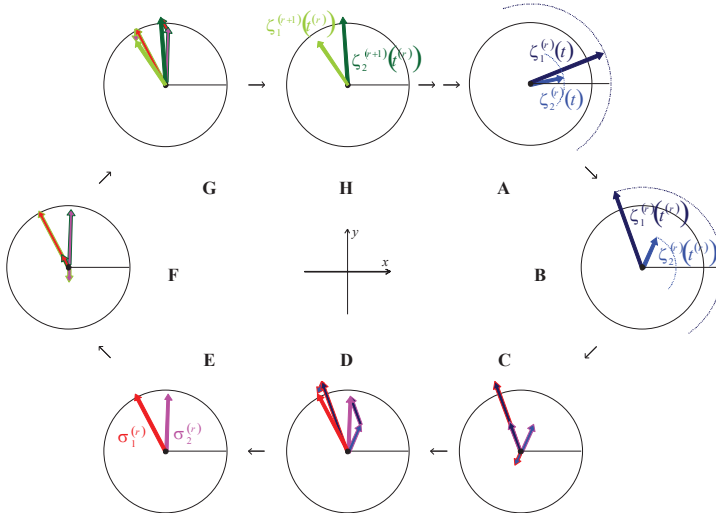
$$\zeta_p^{(r)}(t) = \exp(i\omega_p^{(r)}(t - t^{(r-1)})) \cdot \zeta_p^{(r)}(t^{(r-1)}). \quad (118)$$

A precesszió a következő csere időpontjáig ($t^{(r)}$) folytatódik (9. ábra B rész). Ekkor az eddigi sajátfüggvények kitüntetett szerepe megszűnik, ezért vissza kell térni a bázisfüggvény-reprezentációra. A 117. egyenlet alapján (és $\sigma_e^{(r)}$ jelöli röviden $\sigma_e(t^{(r)})$ -t):

$$\sigma_e^{(r)} = \sum_p c_{ep}^{(r)} \zeta_p^{(r)}(t^{(r)}). \quad (119)$$

1. Táblázat A 8. ábra spektrumainak szimulációjához használt paraméterek.

	$A^{(1)}$	$A^{(2)}$
v_1/Hz	-50	100
v_2/Hz	0	-20
$J_{1,2}/\text{Hz}$	20	10



9. Ábra A számolás menete vektorokkal. (A) A mágnesezettség vektorok precessziója egy időszelvényben. (B) A vektorok helyzete az időszelvény végén ($t^{(r)}$). (C) A sajátvektorok felbontása báziskoherenciacíkra. (D) Az azonos báziskoherenciacíjú vektorok összeadása. (E) A $t^{(r)}$ időpontban jelenlévő bázisvektorok. Ebben a reprezentációban történik a magcsere. (F) A magcsere után a báziskoherenciací felbontása. (G) Az azonos koherenciacíjú vektorok összeadása. (H) A két új sajátvektor megkezdte precesszióját $t^{(r+1)}$ -ben.

Ezt az átváltást mutatja részletesen az 9. ábra C – E része: a $\zeta_p^{(r)}$ vektorokat a $c_{ep}^{(r)}$ együtthatók alapján felbontjuk (C), majd az azonos báziskoherenciacíhoz tartozókat megfelelőeket összeadva (D) megkapjuk az eredő $\sigma_e^{(r)}$ vektorokat (E). Ezek a vektorok a csere-folyamat során nem változnak meg, így a magcseret ebben a reprezentációban értelmez-zük. A következő időszelvény szimulációja ebből az állapottól folytatható.

Az $r+1$ -dik időszelvényben a σ vektorokról ismét át kell térni a sajátfüggvény-reprezentációra, de már az $r+1$ időszelvényben aktuális konformer szerint, azaz a 117. egyenlet alapján:

$$\zeta_p^{(r+1)}(t^{(r)}) = \sum_p c_{ep}^{(r+1)} \sigma_e^{(r)}. \quad (120)$$

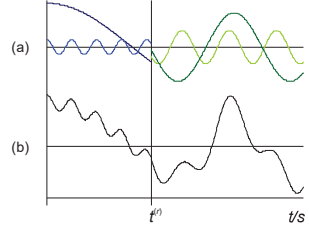
Ezt az időszelvény eleji konverziót a 9. ábra F – H része részletezi, ami formailag hasonló a B – D átváltáshoz. Ezután a precesszió folytatódik (ismét A és B részek következ-nek, de a két vektor a csere miatt megváltozott), immár az új konformernek megfelelő frekvenciákkal (a 118. egyenletben r értéke eggyel nőtt).

Az ilyen módon értelmezett és a 9. ábrán bemutatott vektormodell csak a sűrűségmátrix elemeinek változását szemlélteti, a detektált jelet nem. A f_{id} a sajátfüggvény-reprezentációjú sűrűségmátrix aktuális elemeiből (azaz a $\zeta_p^{(r)}(t)$ vektorból) kaphatjuk meg:

$$f_{id}(t) = \sum_{p=1}^2 a_p^{(r)} \zeta_p^{(r)}(t), \quad (121)$$

ahol $a_p^{(r)}$ ($p = 1, 2$) az egyes jelek komplex amplitúdója (93. egyenlet). Ezt a detektált jelet mutatja

a 10. ábra az r -edik csere környékén. Egymagos esetben az $o_p^{(r)} = 1$ minden r -re (p értéke csak 1 lehet), így a vektor rögtön a detektált jelet adja meg.



10. Ábra A f_{id} kiszámolása egy csere környékén. (a) Az egyes sajátvektoroktól származó görbék valós része. (b) A fentiek szuperpozíciójával kapott teljes f_{id} .

4.6.2. A populációkról

Az előző fejezetben a bázis- és sajátállapotok viselkedését és sűrűségmátrixszal való kapcsolatát vizsgáltuk. Ebben a fejezetben ez egyes állapotoknak a cserék és a precesszió szakaszaiban bekövetkező populációváltozásáról lesz szó.

4.6.2.1. Spinfüggvények viselkedése erősen csatolt spinrendszerben

Első lépésként a spinfüggvények viselkedését vizsgáljuk a spinhalmaz szorzatfüggvényeinek terében (11. ábra). Ebben a vektortérben egy pillanatszerű csere során a spinhalmaz φ_a szorzatfüggvénnyel jellemzett spinállapota megmarad, ilyen értelemben a szorzatfüggvények populációja mikroszkopikusan is állandó. A precesszió során viszont az adott spinhalmaz bázisállapota megváltozhat, így ekkor a φ_a szorzatfüggvények populációja is változik. Ezzel szemben a $\psi_k^{(r)}$ sajátfüggvények konformerfüggők, így azok a csere pillanatában eltűnnek, viszont az r -edik időszelvénybeli precesszió során populációjuk állandó.

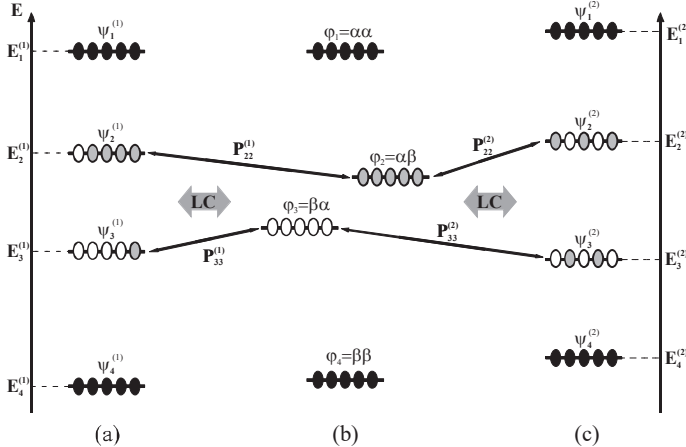
A két reprezentáció közötti átváltást az teszi lehetővé, hogy a $\psi_k^{(r)}$ sajátfüggvények felírhatók a φ_a bázisfüggvények lineáris kombinációjaként (82. egyenlet) és viszont (84. egyenlet). A átváltási lépések matematikája mellett ezek az együtthatók a modellben annak

a valószínűségét adják meg, hogy az r -edik időszelvényben egy φ_a vektor milyen $(P_{ak}^{(r)})$ valószínűséggel van a $\psi_k^{(r)}$ sajátállapotban, konkrétan:

$$P_{ak}^{(r)} = (\mu_{ak}^{(r)})^2. \quad (122)$$

A 11. ábra a spinhalmazok egyes bázis- és sajátállapotok közötti vándorlását mutatja be a detektálási idő alatt különböző eseményeknél. A cserék időpontjában az a meghatározó, hogy egy adott spinhalmaz melyik bázisvektornak megfelelő állapotban van. Ezzel szemben a precesszió ideje alatt (azaz amikor nem történik cserefolyamat) az aktuális sajátvektort vesszük figyelembe. A spinhalmazok tiszta bázisállapotai (a szürke $\alpha\beta$ és fehér $\beta\alpha$ állapotok a 11. ábrán) csatolt rendszerekben kevert sajátállapotokhoz vezethetnek, a keveredés mértékét a 122. egyenletben megadott valószínűségek szabják meg az egyes konformerekre specifikusan. Például a 11. ábrán a $\psi_2^{(1)}$ sajátállapotban a $\beta\alpha$ és $\alpha\beta$ állapotok 1:4 arányban keverednek, míg a $\psi_2^{(2)}$ állapotban 2:3 arányban. A saját- és bázisállapotok közötti átváltás bármely időpillanatban megtehető, de csak a csere pillanatában van jelentősége.

A szimuláció során spinhalmazok sokaságát vizsgáljuk. Ennek megfelelően az egyes



11. Ábra A vektormodell kiterjesztése egy AB spinrendszer esetén. Az egyedi spinhalmazok tiszta bázisállapotai (fekete $\alpha\alpha$ és $\beta\beta$, szürke $\alpha\beta$ és fehér $\beta\alpha$ állapotok) csatolt rendszerekben kevert sajátállapotokhoz vezethetnek. (a) Az első konformer energiaszintjei: sajátállapotok egzakt energiával. (b) Bázisállapotok a csere pillanatában (ezekre nem vonatkozik az energiaskála) (c) Sajátállapotok a csere után. LC a sajátállapotok és bázisállapotok közötti transzformációt jelöli.

állapotok tartózkodási valószínűsége a nagy számok törvénye szerint átváltható relatív populációra, azaz a φ_a bázisállapotból a $\psi_l^{(r)}$ sajátállapotba az összes ott lévő spinhalmaznak a $P_{al}^{(r)}$ valószínűséggel arányos része kerül.

4.6.2.2. Az egykvantum-koherenciák viselkedése

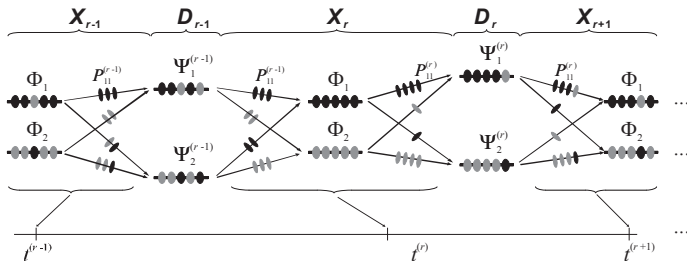
Az előző részben áttekintettük a bázis- illetve sajátfüggvények viselkedését a precesszió illetve a cserefolyamatok során, amely alapján kiszámolható az egyes energiaszintek populációja a detektálás alatt. A *fid* szimulációjához azonban nem a Hilbert-térbeli állapotok, hanem az egykvantum-koherenciák viselkedését célszerű vizsgálni. Ezek a koherenciák két állapot együttes viselkedéséből alakulnak ki. A spinállapotokhoz hasonlóan itt is lineáris kapcsolat van az egyes konformerek sajátfüggvényei (amelyeknek karakterisztikus frekvenciája is van) és báziskoherenciái között. Adott $\Psi_p^{(r)}$ koherencia ($\psi_k^{(r)} \rightarrow \psi_j^{(r)}$ átmenet) felírható a Φ_e báziskoherenciák ($\varphi_a \rightarrow \varphi_b$ átmenetek) lineáris kombinációjaként (90. egyenlet). Az r -edik időszelvényen a Φ_e báziskoherencia esetén a $\Psi_p^{(r)}$ koherencia $P_{ep}^{(r)}$ előfordulási valószínűsége a 122. egyenlethez hasonlóan felírható:

$$P_{ep}^{(r)} = (c_{ep}^{(r)})^2, \quad (123)$$

Ez a valószínűség a 94. egyenlet alapján megegyezik a $\psi_k^{(r)}$ és $\psi_l^{(r)}$ sajátállapot együttes előfordulásának valószínűségével az egyszerre előforduló φ_a és φ_b bázisállapotok esetén, azaz

$$P_{ep}^{(r)} = (c_{ep}^{(r)})^2 = (u_{ak}^{(r)})^2 (u_{bj}^{(r)})^2 = P_{ak}^{(r)} P_{bj}^{(r)}. \quad (124)$$

azaz a különböző totálspinkvantumszámú állapotok előfordulása egymástól független.



12. Ábra A koherenciák populációjának változása a precesszió (D), illetve cserék (X) során, az r . csere környékén. A színezés a változások szemléltetésére szolgál.

A 12. ábrán egy AB spinrendszer r -edik csere környéki populációváltozása látható. Az aktuális cserét megelőző $(r-1)$ -edik csere során a Φ_1 és Φ_2 koherenciákat a 91. egyenletben definiált együtthatók szerint felosztjuk a $\Psi_1^{(1)}$ és $\Psi_2^{(1)}$ sajátfüggvények között (X_{r-1} szakasz). Az ezt követő $r-1$ -edik időszelvényben (D_{r-1} rész) a precesszió zavartalan, ennek során a sajátfüggvények populációja nem változik. Az X_r csere során először a $\Psi_1^{(1)}$ és $\Psi_2^{(1)}$ állapotokról újra áttérünk a báziskoherenciákra. A D_{r-1} precesszió során az egyes specieszek „elfelejtik”, hogy melyik báziskoherenciához tartoztak, ezért a precesszió során bármely bázisfüggvény megváltozhat. A csere során a bázisfüggvények megmaradnak, így csak az új (D_r -ben érvényes) sajátfüggvényekre kell a populációjukat szétosztani, és a ciklus kezdődik előlről. Az egyetlen különbség az, hogy az r -edik időszelvényben (D_r szakasz) az ott érvényes konformer paramétereivel kell számolni.

5. fejezet

Gyakorlati megvalósítás: ProMoCS

5.1. A ProMoCS program szerkezete

5.1.1. A számolás algoritmusa

A 4. fejezetben ismertetett elmélet képezi az alapját a ProMoCS (*Propagators & Monte Carlo for DNMR Spectrum Simulation*) dinamikus NMR spektrumokat szimuláló programnak. Ezt a programot Java, illetve C programozási nyelven írtam meg, a két verziót röviden ProMoCS-Java, illetve ProMoCS-C változatnak nevezzük. Ezek mellett készült egy NVidia GPGPU képes videokártyára írt program is (ProMoCS-CUDA), annak részleteivel külön fejezet foglalkozik.

A program tetszőleges számú konformert érintő cserefolyamatokat tud szimulálni, amelyeknél az erősen vagy gyengén csatolt magokból álló spinhalmaz mérete legfeljebb 8–14 $\frac{1}{2}$ spinű mag. A cserefolyamatok lehetnek intra- vagy intermolekulárisak, kölcsönösök és nem kölcsönösök – akár vegyesen is. További feltétel, hogy az egyes konformerek közötti átalakulás pillanatszerű pszeudo-elsőrendű reakcióval történjen.

A három program közös algoritmusa látható a 13. ábrán. Az első lépésben az input fájl alapján meghatározásra kerülnek a szimuláció paraméterei: a szimulált spektrum általános adatai (*sw*: spektrum szélessége, *si*: spektrumpontok száma, *o1p*: spektrum közepe, *aqt*: detektálási idő, *ns*: scanek száma), a spinrendszer statikus (*n*: spinhalmaz mérete, *m*: konformerek száma, *v*: kémiai eltolódások, *J*: csatolási állandók minden konformerben) és dinamikus adatai (ΔH , ΔS : aktiválási paraméterek, *T*: hőmérséklet-adatok). A következő lépés a kinetikus leíráshoz szükséges d_h és κ_{gh} valamint az egyensúlyi relatív koncentrá-

	void run(inputfile)	
Step 1	mode, o1p, ppm, k^* , <i>n</i> , <i>m</i> , <i>ns</i> , <i>sw</i> , <i>si</i> $J[m][n][v], v[m][n], \Delta H[m][m], \Delta S[m][m], T[]$	$\} = \text{getParams}(\text{inputfile});$ bemeneti paraméterek
Step 2	*thread = initThreads(mode, <i>n</i> , <i>m</i> , <i>ns</i> , <i>si</i>);	programszálak inicializálása
Step 3	$\left\{ \begin{array}{l} d[m], K[m], \kappa[m][m] = \text{calcKinetics}(\Delta H[m][m], \Delta S[m][m], T, m); \\ X[m][m] = \text{setKin}(d, K, \kappa); \end{array} \right.$	kinetikai param. számolása
Step 4	ft[sj] = {0,...,0};	ft gyűjtőmb inicializálása
Step 5'	if(C) calcDNMRS(<i>ns</i> , <i>n</i> , <i>m</i> , <i>si</i> , <i>aqt</i> , $v[m \times n]$, $J[m \times n \times n]$, $X[m \times m]$);	C nyelvű elágazás
Loop 1	else for(<i>l</i> = 1; <i>l</i> ≤ <i>n</i> ; <i>l</i> ++)	
Step 5	$A_{\text{temp}}, U_{\text{temp}}, \Lambda_{\text{temp}} = \text{calcNMRstat}(n, l, v_{\text{stat}}, J_{\text{stat}});$	új <i>A</i> , <i>U</i> , Λ mátrixok meghat.
Step 6	$A_l, U_{l-1}, U_l, \Lambda_l, \Lambda_{l-1} = \text{setNMRstat}(A_{\text{temp}}, U_{\text{temp}}, \Lambda_{\text{temp}}, U_l, \Lambda_l, f);$	<i>A</i> , <i>U</i> , Λ mátrixok átmásolása
Step 7	ft += calcSingleSpec();	fid számolása
Step 8	ft = sumResult(<i>si</i> , <i>thread</i>);	eredmények összeadása
Step 9	Y = fourier(k^* , <i>si</i> , <i>ft</i>);	Fourier-transzformáció
Step 10	saveResult(Y, inputfile);	eredmények mentése

13. Ábra A ProMoCS program algoritmusa.

ciók (K_h) meghatározása a cserefolyamatok aktiválási paramétereit alapján és ezek tárolása egy X tömbben (34–36. és 39–40. egyenletek, Step 3) valamint az eredményt tároló \tilde{f} tömb inicializálása (Step 5). Ez után a lépés után válik külön a C nyelvű változat, ami hidat biztosít a ProMoCS-CUDA felé is (Step 5'). A Java nyelvű változatban innentől indul a blokkok (l) szerinti ciklus (Loop 1). Ennek első lépése (Step 5) az egyes konformerek Hamilton mátrixának felépítése (41. egyenlet) és ez alapján az $A_l^{(r)}$, $A_l^{(r)}$, $U_l^{(r)}$ mátrixok meghatározása (42. egyenlet, [142]). Az $l-1$ -dik szint mátrixait az előző függvényhívásban már kiszámoltuk, ezeket csak át kell indexelni (Step 6). Ezután indul a scanek szerinti szimuláció, amelynek algoritmusát a 14. ábra részletezi (calcSingleSpec eljárás, Step 7). Végül az eredmények összegzése (Step 8) és egy Fourier-transzformáció (Step 9) után az eredmények mentésével (Step 10) ér véget a program.

A szimuláció törzstét jelentő calcSingleSpec eljárás (14. ábra) a 4.2–4.4. fejezetekben ismertetett egyenletek alapján egyesével kiszámolja és összeadja a *scanek*et (Loop 1). A *scanek* szerinti ciklus első lépése a kezdeti konformer meghatározása (Step 1, 35. egyenlet) és a sűrűségmátrix felépítése (Step 2). Ezután indul az időszletek szerinti ciklus (Loop 2),

```

calcSingleSpec()
osztályváltozók:  $n, n_{s1}, m, si, aqt, U[m \times n \times n], U_{s1}[m \times n_{s1} \times n_{s1}], A[m \times n \times n_{s1}], \Lambda_s[m \times n], \Lambda_{s1}[m \times n_{s1}]$ 
output:  $\tilde{f}[si]$ 

Loop 1      for( $s = 0; s < ns; s++$ )
Step 1       $h^{(0)} = \text{calcFirstConf}(X, \varepsilon); t^{(0)} = 0;$                                 első konformer
Step 2      for( $j = 0; j < n_s; j++$ ) for( $k = 0; k < l; k++$ )  $\xi[\text{iplus}(j, k)] = 1;$         kiindulási  $\xi$  meghat.
Loop 2      while( $t^{(0)} < aqt$ )
Step 3       $\Delta t^{(0)} = \text{calcSliceLen}(X, \varepsilon);$                                        időszlet hossza
Step 4      {
        for( $b = 0; b < n_s; b++$ ) for( $k = 0; k < n_{s1}; k++$ )
        {
             $\xi_{\text{temp}}[k, b] = \sum_{a=1}^{n_1} U_{s1}[h^{(0)}, k, a] \xi[a, b];$ 
            for( $l = 0; l < n_s; l++$ ) for( $k = 0; k < n_{s1}; k++$ )
            {
                 $\xi[k, l] = \sum_{b=1}^{n_1} \xi_{\text{temp}}[k, b] U[h^{(0)}, b, l];$ 
            }
        }
    }
    {
        for( $i = \text{index}(t^{(0)}), t = 0; t < \Delta t^{(0)}; i++, t += aqt/si$ )
        {
             $\tilde{f}[i] += \sum_{k=1}^{n_1} A[h^{(0)}, l, k] \exp(i(\Lambda[h^{(0)}, l] - \Lambda_{s1}[h^{(0)}, k])(t - t^{(0)})) \cdot \xi[k, l];$ 
        }
    }
    {
        for( $l = 0; l < n_s; l++$ ) for( $k = 0; k < n_{s1}; k++$ )
        {
             $\xi[b, k, l] \times = \exp(i(\Lambda[h^{(0)}, l] - \Lambda_{s1}[h^{(0)}, k]) \cdot \Delta t^{(0)});$ 
        }
    }
    {
        for( $b = 0; b < n_s; b++$ ) for( $k = 0; k < n_{s1}; k++$ )
        {
             $\xi_{\text{temp}}[k, b] = \sum_{a=1}^{n_1} U_{s1}[h^{(0)}, k, a] \xi[a, b];$ 
            for( $l = 0; l < n_s; l++$ ) for( $k = 0; k < n_{s1}; k++$ )
            {
                 $\xi[k, l] = \sum_{b=1}^{n_1} \xi_{\text{temp}}[k, b] U[h^{(0)}, b, l];$ 
            }
        }
    }
Step 8       $h^{(0)}, t^{(0)} = \text{calcConf}(\Delta t^{(0)}, h^{(0)}, X, \varepsilon);$                             következő konformer

```

$\sigma - \xi$ transzformáció
Eq. 80/2.
 jel számolása
Eq. 81
 propagálás
Eq. 79
 $\xi - \sigma$ transzformáció
Eq. 80/1

14. Ábra A spektrumszimuláció magjának részletes algoritmus.

amelynek minden egyes lefutása az adott időszület fid_j -ét adja az eredményhez (fi) valamint propagálja a sűrűségmátrixot az időszület végére. Ebben a ciklusban az időszület hosszának meghatározása (38. egyenlet, Step 3) után a sűrűségmátrixot a szorzatfüggvények bázisáról az aktuális konformer sajátfüggvényeinek bázisára transzformáljuk (80. egyenlet második része). Ebből a mátrixból számoljuk a 81. egyenlet szerinti a $fid^{(r)}$ értékét az összes érintett pontban (Step 5) valamint az időszület végén érvényes sűrűségmátrixot (79. egyenlet, Step 6). Végül ezt a $\xi_j^{(r)}(t^{(r)})$ mátrixot visszatranszformáljuk a szorzatfüggvények bázisára (80. egyenlet első része, Step 7), így megkapjuk $\rho^{(r)}$ -t. Az utolsó lépés a következő időszületben aktuális konformer meghatározása (39. egyenlet, Step 8), innen folytatódik Loop 2 előlről.

5.1.2. Megosztás több processzoron

A ProMoCS spektrumszámoló program statisztikus alapon közelíti a kísérleti spektrumot. Ennek legfontosabb következménye, hogy a szimuláció pontossága (jel/zaj viszonya) a független számolások (*scanek*) számától függ. Az elfogadható minőségű spektrumhoz általában 500 körüli *scan* szükséges, ami nagyobb spinrendszerek és gyors cserefolyamatok esetén több óras vagy napos számolást is igényelhet.

A számolás időigénye csökkenthető, ha azt több gépen vagy processzoron megosztva végezzük. A többszálúsítást a Java nyelv Thread osztályának és a java.rmi csomag objektumainak használatával végeztem.

A programban a calcSingleSpec eljárást lehet párhuzamosan több szálon futtatni. Az eljárás által használt változókat először a szálak saját memóriaterületeikre másolják, erre szolgál a 13. ábrán az initThread metódus (Step 2). A Hamilton-operátor blokkjaitól függő mátrixokat Loop 1-ben a setNMRstat (Step 6) eljárás másolja át. Ez utóbbi lépést a szálak már párhuzamosan hajtják végre, így a program többszálúsított része Step 6–7 a 13. ábrán.

A többmagos processzorokon a program párhuzamos szálakon futhat. Emellett lehetőség van több gépen párhuzamosan futtatni a programot. Ezt a Java RMI (Remote Method Invocation) technológiája segítségével valósítottuk meg. A párhuzamosítás elve megegyezik az egy gépen való többszálúsítással, a különbség csak annyi, hogy a szálak inicializálásához az előre definiált szerveren bejegisztált távoli gépekhez kell kapcsolódnia, majd az egyes metódushívások ezen a kapcsolaton keresztül már ugyanúgy történnek, mintha a szálak ugyanazon a PC-n futnának.

5.2. ProMoCS-CUDA

A Monte Carlo alapú DNMR szimuláció az azonos műveletek többszöri ismétlése és a gyakori mátrixműveletek miatt hatékonyan átirható a GPU-ra. A CUDA kód C nyelvű környezetből hívható a legegyszerűbben, ezért a Java nyelvű ProMoCS programhoz először egy C nyelvű algoritmusrészt írtunk, ezen keresztül lehet a videokártyán inicializálni a szimulációt. A C nyelvű algoritmus vázlatát mutatja a 15. ábra.

A szimuláció időigényes része a *fid* kiszámolása, ezért ez kerül át a GPU-ra. A mátrixok diagonalizálása és a Fourier-transzformáció nem igényel sok időt, ezért ezt a két lépést továbbra is a CPU végzi.

Az implementáció alapja a *scanek*, azaz a 52. egyenletben az *s* szerinti szétválasztás. Az egyes *scanek*hez szükséges számolásokat néhány szál (legfeljebb egy blokk) végzi. Az egy *scanen* dolgozó szálak száma a sűrűségmátrix méretétől függ. A blokkok méretkorlátja miatt két külön algoritmus van a blokkok dimenziójánál kisebb és a nagyobb mátrixokra.

A C nyelvű eljárás a memóriaterületek allokálásával és a véletlenszám generátor inicializálásával indul (Step 1). Ezután már a sűrűségmátrix blokkjai (*l*) szerinti számolás következik. Ennek során a Java nyelvű változat *calcDNMRstat* metódusának megfelelő eljárás a 41. egyenlet és a bemeneti adatok alapján felépíti és diagonalizálja az adott *l*-hez tartozó

```

void calcDNMRs(ns, n, m, si, aqt, v[m×n], J[m×n×n], X[m×m])
output: fid[si]
Step 1 { cudaMalloc(...); malloc(...);           X, ft, A, U, Λ tömbök allokálása
        si = calcSize();                         függvényhívásban számolt scanek
        rnd = initRnd();                         véletlenszámgenerátor inicializálása
Loop 1  for(l = 1; l ≤ n; l++)
Step 2    Al, Ul-1:l, Up, Λp, Λp-1:p, np, np-1 = calcDNMRstat(n, vstat, Jstat);   A, U, Λ mátrixok és méreteik meghat.
        if(cuda)
Step 3      g, dg, dp = calcDim(l);           blokkok és grid mérete
Step 4      ft = cudaCopy(Ap, Ul-1:l, Up, Λp, Λp-1:p);       mátrixok másolása, ft inicializálása
        if(small)
Step 5a      for(s = 0; s < ns; s += si)
                calcSmall<<< g, dg, dp >>>(...);           kernelhívás kis mátrixokra
        else
Step 5b      for(s = 0; s < ns; s += si)
                calcLarge<<< g, dg, dp >>>(...);           kernelhívás nagy mátrixokra
Step 6      ft += sum<<< 1, g >>>(ft, si);           blokkok eredményeinek összeadása
Step 6'     else fid += calcSingleSpec();             C nyelvű szimuláció
Step 7      if(cuda) fid = cudaCopyResult(si);        eredmény visszaszámolása
Step 8      cudaFree(...); free(Ap, Ul-1:l, Up, Λp, Λp-1:p);   memória felszabadítása

```

15. Ábra A C nyelvű szimuláció algoritmusának vázlata.

a $H_l^{(r)}$ mátrixokat, ennek eredményei az $A_l^{(r)}$, $U_l^{(r)}$ és $\Lambda_l^{(r)}$ mátrixok (Step 2). A program a CUDA-ban definiált *grid* pontos paramétereinek meghatározása (Step 3) után az 2. lépés eredményeként kapott mátrixokat a videokártyára másolja (Step 4). A kártyán történő szimuláció (Step 5) algoritmus a különböző kis és nagy mátrixokra, ezek részleteit a következő két fejezet tárgyalja. A *kernel*en történő számolás során minden blokk egyetlen eredményvektort ad vissza, amely összegzi az adott blokk által számolt *scan*eket. A számolás végén ezeket a tömböket még a kártyán elemenként összegezzük (Step 6), és utána az eredményt visszamásoljuk a PC RAM-jába (Step 7) majd a program visszatér az eredeti ProMoCS program menetébe.

5.2.1. Nagy sűrűségmátrixok

A nagy sűrűségmátrixok az öt magnál nagyobb spinrendszerek $1 < l < n$ blokkjai. Ekkor a szimulált sűrűségmátrix-blokkok nagyobbak, mint az egy blokkon belüli szálak maximális száma. Emiatt a nagy mátrixok szorzására ismert modellt alkalmazunk a számolás során [111]. A $\xi^{(r)}$, $A_l^{(r)}$, $U_l^{(r)}$ és $\Lambda_l^{(r)}$ mátrixokat a globális memóriában tároljuk (ξ , A , U , Λ) és mindig csak azokat a részleteit másoljuk a GPGPU megosztott memóriájába (Y_R és Y_l mátrixok), amelyet a blokk szálai éppen használnak (16. ábrán Step 8-ban, valamint a Step 10-ben a 2. és 5. sor). Ezáltal a blokk a sűrűségmátrixot egy kis ($d_x \times d_y = 16 \times 16$) “ablakon” keresztül látja. Ez az ablak a számolás minden egyes lépésében végigfut a teljes mátrixon, így biztosítva, hogy minden egyes blokk pontosan egy spinhalmaz sűrűségmátrixát számolja. (Ez eltérés a klasszikus CUDA mátrixszorzástól.)

A számolás algoritmus a CPU szimulációtól jelentősen eltér. A ProMoCS-C és a ProMoCS-Java `calcSingleSpec` eljárásának fő ciklusa (Loop 2 a 14. ábrán) az időszeletek szerint fut, és minden egyes lépésben kiszámolja az aktuális időszület fídjét. Ezzel szemben a CUDA nyelvű algoritmus fő ciklusa a *fí*d pontjai szerint halad és minden egyes pont előtt annyi időszeleten propagál át, amennyire szükség van. A kétféle módszer műveletigénye megegyezik, viszont szinkronizáció szempontjából a videokártyán az utóbbi megoldás sze-rencsebb.

Az algoritmus vázlatát a 16. ábra tartalmazza. A kernelhívás paraméterei a konformereket jellemző $A_l^{(r)}$, $U_l^{(r)}$ és $\Lambda_l^{(r)}$ mátrixok, a cserefolyamatokat és az egyensúlyi állapotot leíró X mátrix, a szimulált spektrum és a függvényhívás néhány alapadata (akvizíciós idő: *aqt*, szimulált pontok száma: *si*, konformerek száma: *m*, a véletlen számok gene-

rálásához szükséges ε tömb és a sűrűségmátrix méretei: n_1, n_2), valamint a sűrűségmátrixnak lefoglalt hely pointere (ξ, ξ_{temp}). Első lépésként a szálak meghatározzák azt, hogy hányadik blokkban vannak (Step 1), ami az eredmény és a köztes számítási adatok globális memóriában történő tárolásához szükséges (f_i illetve ξ, ξ_{temp} tömbök hivatkozásánál az első index mindig a blokk sorszáma, b). Ezt követi a szimuláció inicializálása: az első konformer (c_r index) és az első időselet hosszának meghatározása ($\Delta t^{(r)}$, Step 2), majd ez alapján a ξ mátrix feltöltése (Step 3). Ezután indul a *fid* számolása (Loop 1), minden egyes ciklusban a *fid* (f_i tömb) egy pontját határozzuk meg (Step 5–6, 81. egyenlet alapján). A

```

__global__ void calcLarge(...)
globális memória:  $\xi, c_{\text{conv}}, n_j, n_{i+1}, m, si, aqt, \xi[g]$ ;
                   $U_{i+1}[m \times n, x_{n,i}] A[m \times n, x_{n,i}] \Lambda[m \times n, x_{n,i}] \Lambda_{i+1}[m \times n, i]$ 
megosztott memória:  $Y[d_c \times d_c], \Delta t^i, t^i, h^i, X[m \times m]$ 
output:  $ft[g \times si]$ 

Step 1   $b = \text{getPosition}()$ ; szál pozíciója
Step 2   $h^i, \Delta t^i = \text{calcFirstConf}(X, \xi[b]); t^i = 0$ ; első konformer meghat.
Step 3  for( $j = 0; j < n_j; j = j + d_j$ ) for( $k = 0; k < n_{i+1}; k = k + d_c$ ) kiindulási  $\xi$  meghat.
         $\xi[b \cdot j + p_c, k + p_c] = A[h^i \cdot j + p_c, k + p_c];$ 
Loop 1  for( $i = 0, t = 0; i < si; i++, t = aqt[si]$ )
Step 5  {
        for( $j = 0; j < n_j; j = j + d_j$ ) for( $k = 0; k < n_{i+1}; k = k + d_c$ )
        {
             $Y[p_c, p_c] += A[h^i \cdot j + p_c, k + p_c] \times$  jel számolása
             $\times \exp(i(\Lambda[h^i, k + p_c] - \Lambda_{i+1}[h^i, j + p_c])(t - t^i)); \xi[b \cdot j + p_c, k + p_c];$  Eq. 81
        }
Step 6   $ft[j] += \sum_{p_c=0}^{d_c-1} Y[p_c, p_c];$  eredmények összegzése
Loop 2  while( $t^i < t$ )
Step 7  {
        for( $j = 0; j < n_j; j = j + d_j$ ) for( $k = 0; k < n_{i+1}; k = k + d_c$ )
        {
             $\xi[b \cdot j + p_c, k + p_c] \times \exp(i(\Lambda[h^i, k + p_c] - \Lambda_{i+1}[h^i, j + p_c])(t - t^i));$  propagálás
            for( $j = 0; j < n_j; j = j + d_j$ ) for( $k = 0; k < n_{i+1}; k = k + d_c$ ) for( $c = 0; c < n_{i+1}; c = c + d_m$ ) Eq. 79
             $Y_s[p_c, p_c] = U[h^i \cdot j + p_c, c + p_c];$ 
             $Y[p_c, p_c] = \xi[b \cdot c + p_c, k + p_c]; \text{syncthreads}();$ 
             $\xi_{\text{conv}}[b \cdot j + p_c, k + p_c] = \sum_{a=0}^{d_c-1} Y_s[p_c, a] Y[a, p_c]; \text{syncthreads}();$ 
Step 8  for( $j = 0; j < n_j; j = j + d_j$ ) for( $k = 0; k < n_{i+1}; k = k + d_c$ ) for( $c = 0; c < n_{i+1}; c = c + d_m$ )
            {
                 $Y_s[p_c, p_c] = U_{i+1}[h^i, c + p_c, k + p_c];$ 
                 $Y[p_c, p_c] = \xi_{\text{conv}}[b \cdot j + p_c, c + p_c]; \text{syncthreads}();$ 
                 $\xi[b \cdot j + p_c, k + p_c] = \sum_{a=0}^{d_c-1} Y[p_c, a] Y_s[a, p_c]; \text{syncthreads}();$ 
            }
             $h^i, \Delta t^i, t^i = \text{calcConf}(t^i, h^i, X, \xi[b]);$ 
Step 9  for( $j = 0; j < n_j; j = j + d_j$ ) for( $k = 0; k < n_{i+1}; k = k + d_c$ ) for( $c = 0; c < n_{i+1}; c = c + d_m$ )
            {
                 $Y_s[p_c, p_c] = U[h^i \cdot j + p_c, c + p_c];$ 
                 $Y[p_c, p_c] = \xi[b \cdot c + p_c, k + p_c]; \text{syncthreads}();$ 
                 $\xi_{\text{conv}}[b \cdot j + p_c, k + p_c] = \sum_{a=0}^{d_c-1} Y_s[p_c, a] Y[a, p_c]; \text{syncthreads}();$ 
Step 10  for( $j = 0; j < n_j; j = j + d_j$ ) for( $k = 0; k < n_{i+1}; k = k + d_c$ ) for( $c = 0; c < n_{i+1}; c = c + d_m$ )
            {
                 $Y_s[p_c, p_c] = U_{i+1}[h^i, c + p_c, k + p_c];$ 
                 $Y[p_c, p_c] = \xi_{\text{conv}}[b \cdot j + p_c, c + p_c]; \text{syncthreads}();$ 
                 $\xi[b \cdot j + p_c, k + p_c] = \sum_{a=0}^{d_c-1} Y[p_c, a] Y_s[a, p_c]; \text{syncthreads}();$ 
            }
        }
    }

```

16. Ábra A nagy sűrűségmátrixokra írt kernelfüggvény algoritmusának vázlata.

trace-ben szereplő összegzést két részletben végezzük: először a szálak a különböző ablakhelyzetekben kapott tagokat összegzik (Step 5), ezután számolják ki a blokk teljes összegét a 2.2.2.1. fejezetben bemutatott algoritmus szerint (Step 6). A *fid* aktuális pontjának kiszámolása után a következő detektált pont előtti utolsó csereidőpontba kell propagálni a sűrűségmátrixot (Loop 2). Ennek során a ζ mátrixot addig propagáljuk újabb csere időpontokba ($t^{(r)}$) amíg az megelőzi a következő detektálási időpontot (t). Ezt a propagálást a Step 7-10 hajtja végre. Először a ζ mátrixot 79. egyenlet szerint kiszámoljuk a következő csereidőpontban (Step 7). Ezt követi a ζ mátrix transzformálása a 80. egyenlet szerint (Step 8 és 10). Ez már valódi mátrixszorzásokat tartalmaz, ezért minden egyes szorzásnál a blokk által számolt ablak végigfut a teljes mátrixon (j és k indexek), és a szorzáshoz szükséges U és ‘rég’ ζ mátrixrészleteket is blokkonként olvassuk be (c index, Y_R és Y_I). Minden egyes ablak beolvasása és minden egyes mátrixrészlet szorzása után a blokk szálait szinkronizálni kell.

A számolás során a 4. fejezetben gyakran szereplő r indexek nem jelennek meg, az időszelletektől függő változók ($h^{(r)}$, $t^{(r)}$, $\Delta t^{(r)}$, $\xi^{(r)}$) mindig felülírják az előzőt. Ezért a konverzió két lépése között (Step 9) kerül sor a csere utáni „új” konformer meghatározására a 39. egyenlet alapján.

A blokkok és a megosztott memóriában tárolt Y mátrixok mérete minden esetben 16×16 . A 16 általában nem osztója a sűrűségmátrix-blokkok méretének, ezért a számolás során egyes szálak néhány ciklusban nem számolnak (ez az elágazás a 16. ábrán az áttekinthetőség kedvéért nincs feltüntetve). Ezek az alvó szálak elkerülhetőek lennének más blokkméret alkalmazásával, de a CUDA architektúrája lényegesen hatékonyabb 16×16 -os mátrixméret esetén, így még az alvó szálak miatti sebességsökkenést is kompenzálja.

5.2.2. Kis sűrűségmátrixok

A kis spinhalmazokra kifejlesztett módszert azokra a sűrűségmátrix-blokkokra alkalmazhatjuk, ahol a blokk mindkét dimenziója kisebb, mint a maximális blokkméret (16×16). Ez gyakorlatilag az 1–5 magból álló spinhalmazok mátrixait jelenti, valamit a nagyobb spinrendszerek első és utolsó blokkját, amelyek dimenziója $1 \times n$, illetve $n \times 1$. Ilyen kis sűrűségmátrixok esetén a szimuláció memóriáigénye néhány kbyte, ezért minden adatot a megosztott memóriaterületeken lehet tárolni. Ennek megfelelően, a különböző konformerek $A^{(r)}$, $U^{(r)}$, $A^{(r)}$ mátrixait és a kinetikai paramétereket a számolás kezdetén a

blokk szálai a saját memóriaterületükre másolják. Hasonlóképpen itt tárolják az aktuális konformerek indexét ($h^{(r)}$), és a cserefolyamatok időpontjait is ($t^{(r)}$, $\Delta t^{(r)}$).

A CUDA nem hatékony, ha kisméretű blokkokkal dolgozik. Ennek elkerülése érdekében egyetlen szálblokk több kis ζ mátrixot kezel, a megosztott memóriaterületen tárolt Y mátrix ezeket a mátrixokat tartalmazza (így Y egy kompozit sűrűségmátrix). Egy kompozit sűrűségmátrixban több spinhalmaz azonos l értékhez tartozó blokkjai szerepelnek (például egy négymagos spinrendszer esetén $l=1$ értéknél a 16×16 méretű kompozit sűrűségmátrix 16×4 darab 1×4 méretű ζ mátrixblokkot tartalmaz). A kompozit sűrűségmátrix használata miatt a szimuláció nem végezhető el tetszőleges számú spinhalmazra: a *scanszám*nak az egy blokkban lévő ζ mátrixok számának többszörösének kell lenni. A sűrűségmátrix mérete függ l értékétől is, ezért a *scanek* számát az összes ilyen l -re vonatkozó minimális *scanszám* figyelembe vételével kell meghatározni (2. táblázat).

Egy szál pontosan egy mátrixelemet számol ki, ezért a CUDA blokk és az Y mátrix mérete azonos. Az algoritmus működéséhez az kell, hogy a blokk és az Y mátrix mérete egész számú többszörösé legyen a sűrűségmátrixénak. Emiatt az alapértelmezett 16×16 -os blokkot néhány esetben csökkenteni kell (pl. 3 magra 15×15), ami enyhén lassítja a programot.

A kompozit-sűrűségmátrix használata miatt minden egyes szálnak ki kell számolnia és tárolnia kell annak sűrűségmátrix sorszámát (i_m), amelyiknek egy elemét számolja, valamint az elem mátrixban elfoglalt relatív helyzetét is (i_x , i_y). Emellett a csereidőpontokat ($t^{(r)}$, $\Delta t^{(r)}$) és a konformereket ($h^{(r)}$), is tömbben kell tárolni a megosztott memóriaterületen.

A számolási algoritmus vázlatát a 17. ábra mutatja be. Az algoritmus menete hasonló a nagy sűrűségmátrixra bemutatotthoz, csak az egyes lépések megvalósítása különböző. Az első lépés itt is a szál helyzetének meghatározása (Step 1). Ezt követi az adatok bemásolása (Step 2), amely úgy történik, hogy a szálak addig másolják párhuzamosan a mátrixelemeket, amíg a végére nem érnek az A , U , A mátrixoknak. Ezt követi a kiindulási konformer meghatározása (Step 3) és a sűrűségmátrix inicializálása (Step 4). A detektált jel számolása

2. Táblázat A CUDA szimuláció minimális *scanszáma* a különböző méretű spinhalmazok esetén.

spinhalmaz mérete	min. scanszám
1	256
2	128
3	75
4	64
5	30
$9 > n > 5$	32
$n > 8$	16

```

__global__ void calcSmall(...)
globális memória:  $n_s, n_{s+1}, m, si, aqt, X[m \times m], \varepsilon[g \times n]$ ,
 $U[m \times n \times n], U_{s+1}[m \times n_{s+1} \times n_{s+1}], A[m \times n \times n_{s+1}], \Lambda_s[m \times n], \Lambda_{s+1}[m \times n_{s+1}]$ 
megosztott memória:  $Y[d_s \times d_s], U[m \times n \times n], U_{s+1}[m \times n_{s+1} \times n_{s+1}], A[m \times n \times n_{s+1}],$ 
 $\Lambda_s[m \times n], \Lambda_{s+1}[m \times n_{s+1}], \Delta t^0[n], t^0[n], h^0[n], X[m \times m], ok\_all$ 
output:  $ff[g \times s]$ 

Step 1  $b_{i_s, i_p, i_m, p_s, p_y} = \text{getPosition}();$  szál pozíciója
Step 2  $\text{for}(j = 0; j < \text{matrixsize}; j += d_s d_y)$ 
Step 3  $A_{i_s, U_{s+1}, \Lambda_{s+1}, \Lambda_s[\text{index}(j, p_s, p_y)]} = A_{i_s, U_{s+1}, \Lambda_{s+1}, \Lambda_s[\text{globalindex}(j, p_s, p_y)]};$  A, U,  $\Lambda$  másolása
Step 4  $h^0[i_m], \Delta t^0[i_m] = \text{calcFirstConf}(X, \varepsilon[b, i_m]); t^0[i_m] = 0;$  első konformer meghat.
Step 4  $Y[p_s, p_y] = A[h^0[i_m], i_m, i_j];$  kiindulási  $\xi$  meghat.
Loop 1  $\text{for}(j = 0, t = 0; i < si; i++, t += aqt/si, ok\_all = false)$ 
Step 5  $\left\{ \begin{array}{l} ff[j] += \sum_{p_y=0}^{d_y-1} \sum_{p_s=0}^{d_s-1} A[c^0[i_m], i_p, i_a] \cdot \exp(i(\Lambda_s[h^0[i_m], i_j] - \Lambda_{s+1}[h^0[i_m], i_j]) \cdot (t - t^0[i_m])) \times \\ \times Y[p_s, p_y]; \end{array} \right\}$  jel számolása, összege Eq. 81
Loop 2  $\text{while}(!ok\_all)$ 
Step 6  $ok = (t^0[i_m] < t); \quad ok\_all = ok \ \&\& \ ! (t^0[i_m] < t);$  propagálás szükségessége
Step 7  $\left\{ \begin{array}{l} \text{if}(!ok) \ Y[p_s, p_y] = Y[p_s, p_y] \times \\ \times \exp(i(\Lambda_s[h^0[i_m], i_j] - \Lambda_{s+1}[h^0[i_m], i_j]) \cdot \Delta t^0[i_m]); \end{array} \right\}$  propagálás Eq. 79
Step 8  $\left\{ \begin{array}{l} \text{if}(!ok) \ F = \sum_{a=0}^{a-1} U[h^0[i_m], i_p, a] \cdot Y[p_y - i_y + a, p_y]; \text{syncthreads}(); \\ \text{if}(!ok) \ Y[p_s, p_y] = F; \text{syncthreads}(); \end{array} \right\}$   $\xi - \sigma$  transzformáció Eq. 80/1.
Step 9  $h^0[i_m], t^0[i_m], \Delta t^0[i_m] = \text{calcConf}(h^0, t^0, X, \varepsilon[b, i_m]);$  következő konformer
Step 10  $\left\{ \begin{array}{l} \text{if}(!ok) \ F = \sum_{a=0}^{a-1} U[h^0[i_m], i_p, a] \cdot Y[p_y - i_y + a, p_y]; \text{syncthreads}(); \\ \text{if}(!ok) \ Y[p_s, p_y] = F; \text{syncthreads}(); \\ \text{if}(!ok) \ F = \sum_{a=0}^{a-1} Y[p_y - i_y + a] \cdot U_{s+1}[h^0[i_m], a, i_j]; \text{syncthreads}(); \\ \text{if}(!ok) \ Y[p_s, p_y] = F; \text{syncthreads}(); \end{array} \right\}$   $\sigma - \xi$  transzformáció Eq. 80/2.

```

17. Ábra A kis sűrűségmátrixokra irt kernelfüggvény algoritmusának vázlata.

során (Step 5) meghatározzuk a blokk által számolt összes jel összegét. A detektált pont kiszámolása után következik a propagálási ciklus (Loop 2). A ciklus folytatásának feltétele az, hogy az összes a blokk által számolt spinhalmaz között van-e olyan, amit még propagálni kell (csak ebben az esetben lesz az *ok_all* logikai változó igaz). Ezen felül, a cikluson belül az adott szál minden egyes lépésnél eldönti a lépés szükségességét (*ok* változó, Step 6). Erre a bonyodalomra azért van szükség, mert a cikluson belül néhány szinkronizálásra is szükség van, amit minden szála végre kell hajtani. A propagálás (Step 7) után következő konverzió (Step 8 és 10) során nincs szükség (és hely) egy átmeneti tároló mátrixra a megosztott memóriában. Ehelyett egy regiszterváltozóban tároljuk az eredményt (*F*), amelyet a szorzás után minden egyes szál a megfelelő helyre visszamásol. A másolás előtt és után ebben az esetben is szükség van a szálak szinkronizációjára.

5.3. Teszteredmények

5.3.1. Tesztelési módszer

A ProMoCS program futási idejét két szempontrendszer szerint vizsgáljuk meg. Egyrészt az új program egyes változatainak sebességét viszonyítjuk egymáshoz és az átlagsűrűségmátrix kiszámításán alapuló programok egy képviselőjéhez (MEXICO), másrészt azt mutatjuk be, hogy az egyes paraméterek milyen módon befolyásolják a számolás sebességét.

A teszteléshez elsőként hipotetikus spinrendszerek spektrumait szimuláltuk. A spinhalmazok 1–8 egymással csatolt magot tartalmaznak. Az egyes spinhalmazoknak 2–4 konformerük van, ezek között nem kölcsönös cserefolyamatot tételeztünk fel (a konkrét adatokat ld. a F.1. függelékben). Így a ProMoCS programmal összesen 24 spinrendszer spektrumsorozatát szimuláltuk. A számolások során a rezonanciafrekvenciát 200 MHz-nek vettük. A MEXICO programban csak kétoldali cserefolyamat szimulációjára van lehetőség, és azokban az esetekben, ahol a két konformer egyensúlyi koncentrációja nem egyezik meg, a cserefolyamat mátrixát rosszul építi fel. Ezen felül a nyolcmagos spinhalmaz esetében a futási idő túlságosan hosszú lett volna. A felsorolt okok miatt a MEXICO programmal csak a futási idők összevetésére van lehetőség kétoldali cserében részt vevő legfeljebb hétmagos spinhalmazok esetén.

Minden spinrendszerhez öt különböző cseresebességnél számoltuk ki a spektrumokat. Az egyes konformerek közötti átalakulások legalacsonyabb hőmérsékletéhez (T_1) tartozó sebességi együtthatóit tartalmazza az F.1. függelék. A magasabb hőmérsékleteken a sebességi együtthatók a táblázatban feltüntetett értékek tíz- (T_2), száz- (T_3), ezer- (T_4), illetve tízezerszeresei (T_5) voltak.

A programmal összesen öt tesztorozatot számoltunk: a ProMoCS-Java programmal egy, illetve két párhuzamos szálon, a ProMoCS-C programmal egy szálon, valamint a ProMoCS-CUDA programmal kétféle videokártyán. Az időeredmények részletesen a F.2. függelékben találhatóak. A szimulált spektrumokat körülbelül 500 *scan*-ból számoltunk, figyelembe véve a 2. táblázatban megadott megkötéseket (3. táblázat).

A számolásokat Microsoft® Windows XP™ ill. Red Hat® Fedora™ 10 operációs rendszer alatt végeztük (Intel® Core™2 Duo 2.20 GHz processzor, 1 Gb RAM). A

3. Táblázat Scanek száma a teszt szimulációkban.

n	ns
1	512
2	512
3	600
4	512
5	600
6	512
7	512
8	512

CUDA™ számításokat NVIDIA® GeForce™ 8600 GT (4 multiprocesszor, 32 mag, 256 Mb RAM) és GeForce™ GTS 250 (32 processzor, 128 mag, 1 Gb RAM) videokártyákon végeztük.

A tesztek során szimulált spektrumok a véletlen számok hibáján belüli eltéréssel azonosak voltak az összes módszer esetében. A szimulált spektrumok a F.3. függelékben láthatók.

5.3.2. Futási időt meghatározó paraméterek

5.3.2.1. Spinhalmaz mérete

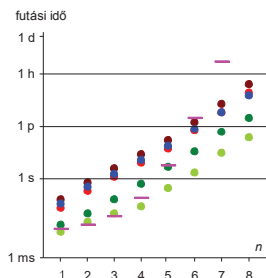
A DNMR spektrumokat szimuláló programok futási idejét alapvetően a mátrixműveletek határozzák meg. A mátrixok méretéről a 4.5.1. fejezetben volt szó, eszerint mind a MEXICO, mind a ProMoCS futási ideje exponenciálisan nő a magok számával, de az előbbi nagyobb alappal. Ezt az elméletet a teszteredmények is alátámasztják (18. ábra).

Kis spinrendszerek esetén (kb. 4–5 magig) a MEXICO lényegesen gyorsabb a ProMoCS programnál, a magok számának növekedésével azonban ez a sorrend még a ProMoCS program leglassabb, Java nyelvű változatára is megfordul.

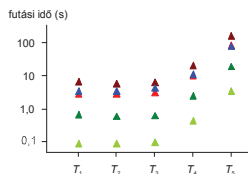
A ProMoCS-CUDA program lényegesen gyorsabb a CPU-s változatoknál, ennek teljes számolási ideje legfeljebb nyolc magot tartalmazó spinhalmazokra a legfeljebb néhány perc, ami az 1–3 magos spinrendszerek kivételével gyorsabb, mint a MEXICO.

5.3.2.2. Sebességi együtthatók

A ProMoCS program futási ideje erősen függ a detektálási idő alatt bekövetkező cserék számától. Ez alapvetően annak köszönhető, hogy



18. Ábra A MEXICO (—) és a ProMoCS program változatainak futási ideje 1–8 magból álló spinhalmazokra kétoldali cserefolyamatokkal: ProMoCS-Java egy (●) ill. két szálon (●●), ProMoCS-C (●) ill. ProMoCS-CUDA két hardveren (● és ●).



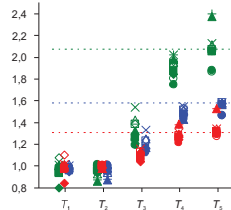
19. Ábra A futási idő változása a sebességi együtthatók (hőmérséklet) növelésével négy mag és három konformer esetén. A különböző színek az egyes programváltozatokat (ProMoCS-Java egy, illetve két szálon: ▲ ill. ▲, ProMoCS-C: ▲ ill. ProMoCS-CUDA: ▲ és ▲) jelölik.

a 79. egyenletben definiált propagálást és a 80. egyenlet szerinti különösen időigényes sűrűségmátrix-transzformációt minden egyes csereidőpontban el kell végezni. A futási idő cseresebességfüggését szemlélteti a 19. ábra. Látható, hogy nagyobb cseresebességnél (T_3 , T_4 és T_5) a számolási idő közel arányos a sebességgel. Alacsonyabb hőmérsékleten (T_1 , T_2) a többi számolási időigénye elnyomja ezt a különbséget. A tapasztalatok szerint a futási idő megugrása nagyjából a koaleszcencia hőmérsékletnek megfelelő sebesség tízszeresénél következik be. Ezzel párhuzamosan a szimulált spektrumban megjelenő jelek is keskenyebbek lesznek és így lehetőség nyílik a *scanszám* (és ezáltal a futási idő) csökkentésére a szimulált spektrumok minőségének romlása nélkül is.

5.3.2.3. Konformerek száma

A ProMoCS program futási ideje elméletileg nem függ a konformerek számától. Ezt mutatja, hogy a 2, 3, illetve 4 konformerrel végzett szimulációk futási ideje a három kisebb reakciósebességnél egymáshoz képest $\pm 10\%$ -os hibahatáron belül volt (20. ábra).

A magasabb hőmérsékleteken a különbséget az okozza, hogy a teszt-rendszerekben a reakciók számának növelésével a konformerek várható élettartama kisebb, így a cserefolyamatok átlagos sebessége nagyobb lesz (4. táblázat). Figyelembe véve, hogy a magasabb hőmérsékleteken a futási idő nagyjából arányos az át-



20. Ábra A különböző számú konformációkkal indított szimulációk futási idejének hányadosa (zöld: négy/kettő; kék: négy/három; piros: három/kettő konformer) öt, illetve hatmagos spinrendszerek esetén (tele, illetve üres szimbólumok) a cseresebesség függvényében. A vízszintes vonal a várható cserefolyamatok számának hányadosát jelöli. A különböző szimbólumok a programváltozatokat (ProMoCS-Java egy, illetve két szálon: ■ ill. ♦, ProMoCS-C: ● ill. ProMoCS-CUDA: ▲ és ×) jelöli.

4. Táblázat A teszt spinrendszerek konformereinek relatív egyensúlyi konformációja, illetve bomlási állandója T_5 cseresebességénél.

h	$m = 2$		$m = 3$		$m = 4$	
	K_h	$d_h \cdot 10^4 / s^{-1}$	K_h	$d_h \cdot 10^4 / s^{-1}$	K_h	$d_h \cdot 10^4 / s^{-1}$
1	0,67	1,00	0,40	2,5	0,33	0,40
2	0,33	2,00	0,20	0,50	0,17	0,29
3			0,40	0,67	0,31	0,40
4					0,19	0,33

lagos cseresebességgel (vízszintes vonalak a 20. ábrán), a futási időben jelentkező eltérést ebben a tartományban nem a konformerek száma okozza.

A MEXICO program esetében a konformerek számára nem lehetett tesztet végezni. Az elméleti alapok alapján a mátrixok mérete nem-kölcsönös cserefolyamat esetén arányos a csereben részt vevő konformerek számával, tehát a futási idő is erősen függ tőle. Kölsönös csere esetén a konformerek száma nem befolyásolja a spinrendszer méretét, így a futási időt sem.

5.3.3. CPU és GPU összehasonlítása

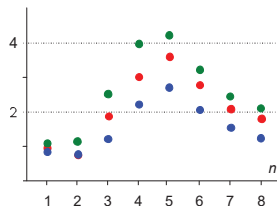
5.3.3.1. Scanek száma

A CPU-n futtatott teszteknel adott spinrendszer és cseresebességek esetén a futási idő egyenesen arányos a szimulált *scanek* számával. Emiatt a többszálú szimuláció futási ideje alapvetően fordítottan arányos a párhuzamosan futó szálak számával, amint ez az egy-, illetve kétszálú ProMoCS-Java szimulációkból is látható. Ezzel szemben a GPGPU speciális architektúrája miatt a CUDA szimulációk ideje nem arányos a *scanek* számával, még a blokkok mérete által szabott korlátokon túl sem.

A jelenség vizsgálata érdekében T_3 hőmérsékleten minden spinrendszer spektrumát szimuláltuk a 3. táblázatban feltüntetett scanszám négyszeresével is. A két futási idő hányadosa a 21. ábrán látható a különböző spinrendszerek esetén. Megállapítható, hogy az elméletileg várható négyszeres sebességcsökkenés csak néhány esetben következett be, az esetek jelentős részében kétszeres idő alatt lefutott a szimuláció.

5.3.3.2. GPGPU hatékonysága

A GPGPU és a CPU alapú program összehasonlításából kiderül, hogy a ProMoCS-CUDA három mag felett legalább ötvenszer, nyolc mag esetén már akár százszor gyorsabb, mint a ProMoCS-Java program és harmincszor gyorsabb, mint a ProMoCS-C változat (22. ábra). Ezek az adatok a lassabb, GeForce 8600 GT videokártya esetében szeré-



21. Ábra A GPU futási idejének aránya s és $4s$ scan esetén 1–8 magos spinrendszerekben két (●), három (●) ill. négy (●) konformerrel. Az alapértelmezett scanszám, s , megegyezik a 3. táblázatban feltüntetettel.

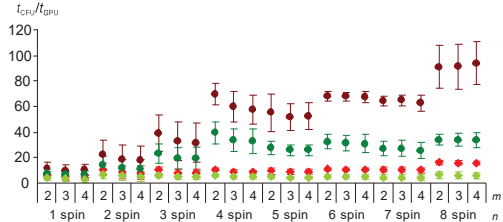
nyebbek. Ez a kártya körülbelül ötször lassabbnak bizonyult, mint a GeForce GTS 250 kártya. Ezek az adatok azonban nem egy-
ségesek a különböző spin-
rendszerek esetében.

A különböző csere-
sebességek esetén a gyor-
sulás mértékének szórása
körülbelül 10% (eltérést
jelző vonalak a 22. ábrán). Az eltérésekben nem mutatkozik tendencia.

A többszálásítás nem annyira hatékony az 1–2 magból álló spinrendszerek esetén, mivel ilyenkor a párhuzamosan futó szálak száma még viszonylag kevés: 512 *scan* esetén 512, illetve 1024 az egy-, illetve kétmagos spinhalmaz esetén. A kártya gyenge kihasználását jelzi az is, hogy a *scanszám* négyszeresére növelése ezekben az esetekben a futási időt gyakorlatilag nem változtatja meg (21. ábra).

A három-, és különösen az ötmagos spinrendszer esetében a sebességnövekedésben enyhe visszaesés tapasztalható a négmagos rendszerhez képest. Ennek az lehet az oka, hogy az első két esetben az egyes blokkok mérete nem az ideális 16×16 -os, hanem 15×15 vagy 10×10 méretű, mivel a sűrűségmátrix dimenziója 3 ($n = 3$) ill. 5 és 10 ($n = 5$).

A nagy sűrűségmátrixok esetében a sebességnövekedés nagyjából azonos mértékű, hatvanszorosa a Java nyelvű programhoz viszonyítva. Ez azzal magyarázható, hogy a futási idő jelentős részében az ideális mátrixmérettel történik a számolás és csak a sűrűségmátrix széléhez érve vannak jelen alvó szálak, ami a számolási időnek nem teszi ki jelentős részét.



22. Ábra A ProMoCS-CUDA program átlagos sebessége a ProMoCS-C (zöld) ill. ProMoCS-Java (vörös szimbólumok) változatához viszonyítva. A GPU teszteket két NVidia videokártyán végeztük: GeForce 8600 GT (● ill. ●) és GeForce GTS 250 (● ill. ●).

6. fejezet

Alkalmazási példák

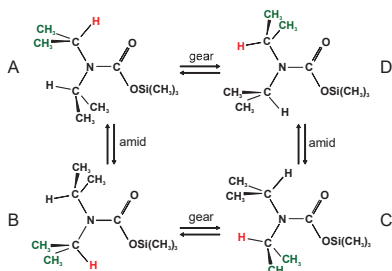
A ProMoCS programot számos valós rendszeren is teszteltük. Ezek közül az N,N-diizopropil-karbaminsav-trimetilszililészter [143], a trimetilszililciklopenta-[7]-fenantrén [144], az N,N-dimetil-para-nitrozo-anilin [123] és egy arany-komplex [145] hőmérséklet-függő spektrumait mutatjuk be.

6.1. Egy nagy spinrendszer

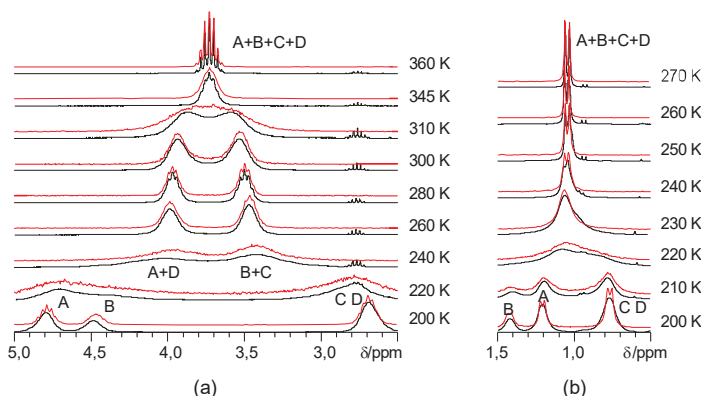
A 23. ábra az N,N-diizopropil-karbaminsav-trimetilszililészter szerkezetét ($^1\text{PrSiC}$) és izomerizációs reakcióit mutatja [143]. A lehetséges reakciók az amid kötés körüli forgás és a két izopropil csoport szimultán forgása a $\text{N}-\text{C}(\text{Pr})$ kötés körül (gear-rotáció).

A kísérleti spektrumok (24.a ábra, Bruker® AVANCE 250 spektrométer, $\text{DMSO}-d_6$) tanúsága szerint

a gear rotáció szobahőmérséklet alatt, 220–240 K között mutat koaleszcenciát az izopropil-csoport metin-protonjának jelén (250 MHz-es készüléken). A 280 K-es spektrumban a karbonilcsoporthoz viszonyított cisz illetve transz helyzetű metin-hidrogének jelei már



23. Ábra Az N,N-diizopropil-karbaminsav-trimetilszililészter ($^1\text{PrSiC}$) szerkezete és izomerizációs reakciói.



24. Ábra Az $^1\text{PrSiC}$ kísérleti (fekete) ill. szimulált (piros) ^1H NMR spektrumának (a) metin (b) metil része különböző hőmérsékleteken.

majdnem kiélesednek, a hőmérséklet további növelésével azonban beindul az amidkötés körüli rotáció, ami ismét a jelek szélesedéséhez vezet. Az újabb koaleszcencia 310 K környékén van, majd megindul a jelek élesedése, ami 360 K-n már egyetlen szeptettet eredményez.

A metil-protonok tartományán (24.b ábra) a két folyamathoz tartozó jelalakváltozás nem válik ilyen élesen ketté a kisebb eltolódáskülönbség miatt. A legszélesebb jeleket 220 K-en tapasztaljuk és 270 K-es spektrumban már teljesen kiélesedett csúcsokat láthatunk.

Az $^1\text{PrSiC}$ molekula cserefolyamatait már korábban tanulmányozták dinamikus NMR spektroszkópiával és a spektrumok szimulációját, illetve a paraméterek finomítását is elvégezték. A metin-proton szimulációjához, a memóriagény csökkentése érdekében, szimmetriát is kihasználó egyszerűsítéseket is használtak [132]. A paraméterek illesztésével kapott termodinamikai adatok a következők voltak: $\Delta S_{\text{amid}}^{++} = 15 \text{ Jmol}^{-1}\text{K}^{-1}$ és $\Delta H_{\text{amid}}^{++} = 68,6 \text{ kJ/mol}$ az amid, illetve $\Delta S_{\text{gear}}^{++} = 14 \text{ Jmol}^{-1}\text{K}^{-1}$ és $\Delta H_{\text{amid}}^{++} = 42,9 \text{ kJ/mol}$ a gear-forgásra.

A ProMoCS programmal a szimulációt a metil-protonok szimmetriájának kihasználása nélkül végeztük. A molekula két izopropil csoportja által alkotott spinrendszer egymástól független, így a szimulált spinhalmazt egyetlen izopropil-csoport hét hidrogénje alkotja (ld. a pirossal jelölt H és a zöld CH_3 a 23. ábrán). Ebben az esetben négyféle konformerrel kell számolni, a 24. ábrán az ezekhez tartozó jeleket A, B, C, D jelöli.

A szimulációhoz használt spektrumparamétereket az 5. táblázat tartalmazza. A kémiai eltolódás és a csatolási állandó kismértékű hőmérsékletfüggését a szimulációkban elhanyagoltuk. A spektrum metin és metil tartományát a jobb felbontás érdeké-

5. Táblázat Az $^1\text{PrSiC}$ szimulációjához használt spektroszkópiai paraméterek.

	A	B	C	D
$\nu_{\text{Pr}}/\text{ppm}$	4,77	4,62	2,73	2,72
$\nu_{\text{Me}}/\text{ppm}$	1,42	1,21	0,77	0,77
J/Hz	7,0	7,0	7,0	7,0

6. Táblázat Az $^1\text{PrSiC}$ DNMR spektrumainak szimulációjához használt sebességi együtthatók a különböző hőmérsékleteken.

T/K	$k_{\text{amid}}/\text{s}^{-1}$	$k_{\text{gear}}/\text{s}^{-1}$
200	$3,6 \cdot 10^{-5}$	140
210	$2,3 \cdot 10^{-4}$	503
220	$1,4 \cdot 10^{-3}$	$1,6 \cdot 10^3$
230	$7,7 \cdot 10^{-3}$	$4,7 \cdot 10^3$
240	$3,6 \cdot 10^{-2}$	$1,2 \cdot 10^4$
250	0,15	$3,0 \cdot 10^4$
260	0,54	$7,0 \cdot 10^4$
270	1,8	$1,5 \cdot 10^5$
280	5,6	$3,1 \cdot 10^5$
300	43,1	$1,1 \cdot 10^6$
310	108	$2,0 \cdot 10^6$
345	$1,8 \cdot 10^3$	$1,2 \cdot 10^7$
360	$5,1 \cdot 10^3$	$2,4 \cdot 10^7$

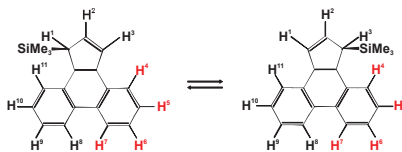
ben külön szimuláltuk. A spektrumokat 512 ponttal, 100 *scan*-ból számoltuk. Az ezekből az adatokból számolt sebességi együtthatókat a 6. táblázat mutatja be. A 24. ábrán látható, hogy a szimulált piros görbék jól közelítik a fekete kísérleti spektrumokat.

6.2. Egy erősen csatolt spinrendszer

A trimetilsziliciklopenta-[1]-fenantrén ($\text{Me}_3\text{Si}^{\text{C}}\text{Pph}$) szerkezete az 25. ábrán látható. A vizsgált folyamat a trimetilszilil-csoport vándorlása C^1 -ről C^3 -ra. Ezt a kölcsönös cserefolyamatot már korábban vizsgálták dinamikus NMR és EXSY kísérletekkel és a mért spektrumok paramétereinek illesztéséből a $\Delta H^{++} = 63,6 \text{ kJ/mol}$ és $\Delta S^{++} = 34,3 \text{ Jmol}^{-1}\text{K}^{-1}$ termodinamikai adatokat kapták. A korábbi mérések során nem sikerült kimutatni azt a nagyfokú konjugációt tartalmazó átmeneti állapotot, amelyben a trimetilszilil-csoport a C^2 szénatomon helyezkedik el, de számítógépes szimulációkkal bizonyították, hogy ennek az átmeneti állapotnak jelentős szerepe van az átalakulás energiájátának csökkentésében [144].

A ProMoCS programmal a nyolc „aromás” proton spektrumát (H^4 – H^{11}) szimuláltuk öt különböző hőmérsékleten (26. ábra). A számoláshoz használt kémiai eltolódás és csatolási állandó értékeket a 7. táblázat tartalmazza.

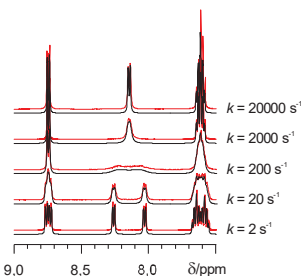
A 25. ábrán pirossal jelölt H^4 – H^7 spinrendszer magjai nem csatolnak skalárisan a H^8 – H^{11} protonokhoz sem, ezért a rendszer



25. Ábra A $\text{Me}_3\text{Si}^{\text{C}}\text{Pph}$ szerkezete és a magok számozása.

7. Táblázat A $\text{Me}_3\text{Si}^{\text{C}}\text{Pph}$ hőmérsékletfüggő spektrumainak szimulációjához használt kémiai eltolódás (δ) és csatolási állandó (J) értékek.

	#1	#2
δ_4/ppm	8,03	8,26
δ_5/ppm	7,60	7,66
δ_6/ppm	7,57	7,63
δ_7/ppm	8,73	8,76
J_{45}/Hz	8,3	8,3
J_{46}/Hz	1,8	1,5
J_{47}/Hz	0,6	0,5
J_{56}/Hz	6,7	6,8
J_{57}/Hz	1,8	1,5



26. Ábra A $\text{Me}_3\text{Si}^{\text{C}}\text{Pph}$ MEXICO (fekete) ill. ProMoCS (piros) programmal szimulált hőmérsékletfüggő spektrumai.

spektrumai leírhatók egy négymagos spinhalmazzal (25. ábrán pirossal jelölve), amelynek két konformere van (amikor a trimetilszilil-csoport egyes – #1 –, illetve hármas pozícióban van – #2).

A szimulált spektrumokat 1000 *scan*-ból számoltuk, 1024 spektrumponttal. A spektrométer rezonanciafrekvenciája 500 MHz volt.

A 26. ábrán látható, hogy a ProMoCS programmal szimulált piros görbék jól illeszkednek a fekete referenciaspektrumokhoz, amelyeket a MEXICO programmal számoltunk.

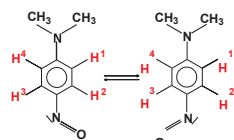
6.3. Csatolt spinrendszer kölcsönös cserével

Az *N,N*-dimetil-para-nitrozo-anilin (NAB) szerkezete a 27. ábrán látható. Ebben a molekulában a nitrozo-csoport szín-anti-izomerizációs átalakulásának aktiválási energiája viszonylag alacsony, ^1H NMR spektrumában szobahőmérséklet környékén koaleszcencia tapasztalható. A nitrozo-csoport közelsége miatt a H^2 proton eltolódása lényegesen nagyobb, mint a másik három aromás magé, ezért tulajdonképpen két koaleszcencia is látható a spektrumokban: az alsó 273 K környékén, amikor a $\text{H}^1\text{--H}^4$ csere koaleszkál és a felső 303 K-n, ahol a $\text{H}^2\text{--H}^3$ páros olvad össze. 303 K-en az első csere már éles jeleket ad. A két magpár jele nem független egymástól, a csatolások miatt egymást is befolyásolják.

Ezt a cserefolyamatot is tanulmányozták már korábban és spektrumillesztéssel meghatározták a kinetikai és a spektroszkópiai paramétereket [123].

A ^1H NMR spektrum aromás részének szimulációjához a négy aromás protont ($\text{H}^1\text{--H}^4$) kell figyelembe venni, mivel ezek a metil-hidrogénekkel nem csatolnak. Ezen négy mag között skaláris csatolás van, és a C--N(O) kötés körüli rotáció páronként összeköti őket. Így a szimulációban a spinhalmaz négy magból áll, amelynek két, egymással ekvivalens konformere van.

A ProMoCS programmal szimulált spektrumokat a MEXICO programmal végzett szimulációhoz hasonlítjuk. A számolásokhoz használt paramétereket a 8. táblázat tartalmazza. A spekt-

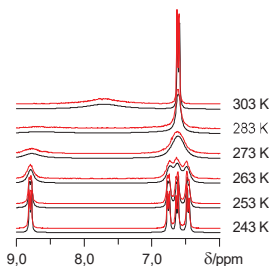


27. Ábra Az *N,N*-dimetil-para-nitrozo-anilin szerkezete és a szimulált protonok számozása.

8. Táblázat Az *N,N*-dimetil-para-nitrozo-anilin hőmérsékletfüggő ^1H NMR spektrumainak szimulációjához használt paraméterek.

δ/ppm	J/Hz
$\delta_1 = 6,76$	$J_{12} = 9,1$
$\delta_2 = 8,79$	$J_{23} = 2,1$
$\delta_3 = 6,63$	$J_{34} = 9,5$
$\delta_4 = 6,47$	$J_{14} = 2,5$

rumokat 1000 *scan*-ból 512 spektrumpontra szimuláltuk (28. ábra). A spektrumközép 7,5 ppm-nél volt, szélessége 3 ppm. A spektrométer rezonanciafrekvenciáját 300 MHz-nek vettük. A 28. ábrán látható, hogy a szimulált piros görbék jól illeszkednek a fekete referencia spektrumokhoz.



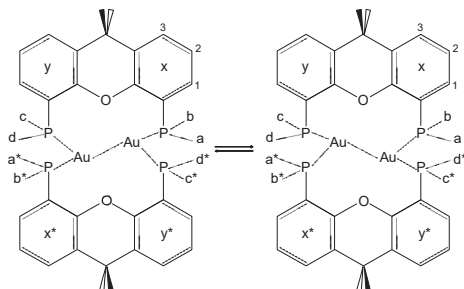
28. Ábra Az *N,N*-dimetil-para-nitrozoanilin MEXICO (fekete) ill. ProMoCS (piros) programmal szimulált hőmérsékletfüggő ^1H NMR spektrumai.

6.4. Xantfoszfát ligandum

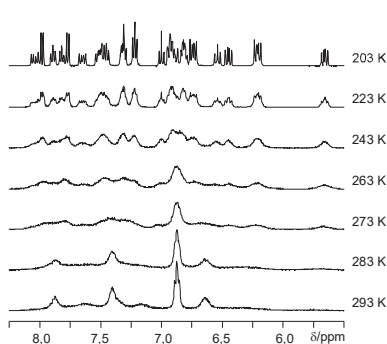
Az arany 9,9-dimetil-4,5-bisz(difenilfoszfino)-xantén ligandummal [145] képzett komplexének (29. ábra) CD_2Cl_2 oldatban felvett spektrumainak tanúsága szerint a ligandumok konformációs mozgása mind a ^{31}P , mind az ^1H spektrumokban szobahőmérséklet környékén mutat koaleszcenciát. A vizsgált változás az aranyatomok körül aszimmetrikusan elhelyezkedő, a ligandumok által alkotott csavart gyűrű konformációs mozgása. A cserefolyamat a királis molekula és enantiomer párja között van, a dinamikus egyensúlyt 1:1 koncentrációarányál éri el. A ligandum egy másik lehetséges konformációs változása a fenilcsoportok forgása. Ha ezt a cserefolyamatot is figyelembe vesszük, a szimulált spektrumok illeszkedése a kísérleti spektrumokhoz nem javul jelentősen, ezért a fenil-gyűrűk forgását a szimuláció során nem vettük figyelembe.

9. Táblázat Az ^1H NMR spektrumok szimulációhoz használt kémiai eltolódások (ppm).

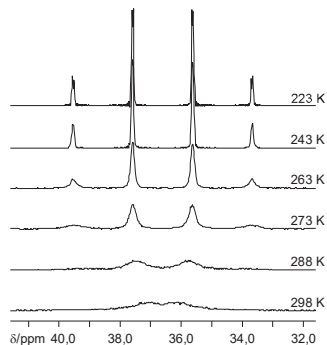
	#1	#2
a ₁	6,21	6,89
a ₂	6,91	6,81
a ₃	7,32	7,49
a ₄	6,91	6,82
a ₅	6,21	7,65
b ₁	6,74	6,54
b ₂	7,22	7,52
b ₃	7,45	7,82
b ₄	7,22	7,89
b ₅	6,74	8,04
x ₁	7,77	7,98
x ₂	7,00	7,31
x ₃	5,66	6,45



29. Ábra Az arany 9,9-dimetil-4,5-bisz(difenilfoszfino)xantén komplex szerkezetének vázlata. Az a, b, c, d egy-egy fenilcsoportot jelöl, amelyek hidrogénjeit sorban 1–5 számozzuk. A *-gal jelölt gyűrűk a csillag nélküliekkel ekvivalensek.



30. Ábra A vizsgált rendszer szimulált hőmérséklet-függő ^1H NMR spektrumai (aromás tartomány).



31. Ábra A vizsgált rendszer szimulált hőmérséklet-függő ^{31}P NMR spektrumai.

A kérdéses cserefolyamat egy kétoldali cserével modellezhető. Az ^1H magok által alkotott spinrendszer független részekre bontható: a két metil-csoportra, a xanténscsoport két aromás gyűrűjének hidrogénjeire (x és y rendszer), valamint a fenilcsoportokra (a , b , c , d -vel jelölve), amelyek mindegyike egy-egy szomszédos ^{31}P atommal is csatol (kivéve a metilcsoportokat). Ezek közül a cserefolyamat összekapcsolja az x -t y -nal, az a gyűrűt d -vel és b -t c -vel. A szimulációt erre a három rendszerre függetlenül lehet elvégezni, és a szimulált spektrumok összege adja az eredményt. Az x - y rendszer egy négymagos (három ^1H mag és a foszfor), míg a b - c , illetve a - d csere egy-egy hatmagos spinhalmaz (a fenilcsoport öt ^1H magja és a csatoló foszfor) kétoldali cserefolyamatával írható le. A szimulált spektrumokban a foszfor jele természetesen nem jelenik meg, csak a csatolása látszik.

A szimulációhoz használt paramétereket a 9. táblázat tartalmazza (Ref. [146] alapján). A spektrumpontok száma 1024, a *scans*szám 512 volt. A spektrométer rezonanciafrekvenciája a kísérletivel azonos 400 MHz volt. A szimulált spektrum aromás részét mutatja a 30. ábra. A szimulált görbék jól illeszkednek a kísérleti spektrumokhoz [146].

A ^{31}P spektrum szimulációjánál az egymással csatolt foszformagokat vesszük figyelembe. A kísérleti spektrumoknál ^1H lecsatolást alkalmaztak, így a protonokkal a szimulációban sem kell foglalkozni. A ^{31}P magok $\text{AA}'\text{BB}'$ spinrendszert alkotnak, a cserefolyamat során az A és a B ill. az A' és B' magok cserélnek

10. Táblázat Az ^{31}P NMR spektrumok szimulációhoz használt kémiai eltolódások δ (ppm) és csatolási állandók J (Hz).

δ_A	37,1 ppm
δ_B	33,7 ppm
$J_{AA'}$	0,7 Hz
$J_{BB'}$	3,3 Hz
J_{AB}	314 Hz
$J_{AB'}$	8,6 Hz

helyet. Így a modell egy két konformerrel rendelkező négymagos spinhalmazt tartalmaz.

A szimulációhoz használt paramétereket a 10. táblázat tartalmazza. A spektrumpon-
tok száma 512, a *scanszám* 576 volt. A spektrométer rezonanciafrekvenciája a kísérletivel
azonos 161,4 MHz volt. A szimulált spektrumokat a 31. ábra mutatja be, amelyek ebben az
esetben is illeszkednek a kísérleti spektrumokhoz [146].

Összefoglalás

Általánosan elterjedt vélekedés a dinamikus NMR spektrumok szimulációjának elméletével foglalkozók körében, hogy a teljes – cserefolyamatokra és skaláris csatolásra zárt – spinrendszer egykvantumátmeneteinek terében a Hamilton-operátor és a cserefolyamatokat leíró szuperoperátor mátrixának összegét kell felépíteni és diagonalizálni ahhoz, hogy megkaphassuk a spektrumot. Eredményeim azt mutatják, hogy ennek nem kell így lennie. A teljes spinrendszer egykvantumátmeneteinek figyelembe vétele csak a szimulációs tér egy olyan bővítése, ami azért szükséges, mert a cserefolyamatok általában nem írhatók le egy spinhalmaz átmeneteinek terében.

PhD munkám során megmutattam, hogy a cserefolyamatok és a skaláris csatolás hatásai szétválaszthatók a szimulációban. A spinek kölcsönhatásait kvantummechanikai, míg a dinamikus jelenséget statisztikus formalizmussal lehet leírni. Ez utóbbit kinetikus Monte Carlo szimulációval lehet egyszerűen kezelni, habár megvan a lehetőség az általános, véletlen számokat nem tartalmazó összefüggés kiszámítására is.

A levezetett képletek lehetőséget adnak az egymagos vektormodell kiterjesztésére is. Így a modellben a dinamikus folyamatok csatolt spinrendszerekben is értelmezhetők. A legfontosabb pont a bázis- és sajátátmenetek közötti átváltás értelmezése a cserék és az egyszerű precesszió során. Ezek a folyamatok elsősorban a koherenciák populációjának változásával függnek össze, és nem írhatók le egyszerűen az egyes bázis-, illetve sajátállapotok közötti kapcsolat alapján. Ennek ellenére, a matematikai leírás a Hilbert-térben is teljes, nem szükséges az átmenetek lényegesen nagyobb dimenziójú terében számolni.

A matematikai képletek alapján megírt program (ProMoCS) a korábbi formalizmus alapján szimulált spektrumokkal megegyező eredményt adott minden tesztrendszer esetében. A ProMoCS program legfontosabb előnye, hogy lényegesen kisebb mátrixokkal számol, mint a korábbi DNMR szimulációs programok. Könnyen és hatékonyan többszálúsítható akár többmagos processzorokon, akár programozható grafikus kártyák egyedi lehetőségeit kihasználva. A videokártyára CUDA segítségével implementált program 30–100-szor gyorsabb, mint az eredeti változat.

A ProMoCS program futási ideje exponenciálisan függ a spinhalmaz méretétől és arányos cserefolyamatok sebességi együtthatójával. Független viszont a konformerek számától, ami miatt elsősorban kis spinhalmazok többoldali cserefolyamatainak szimulációjára hatékony.

Abstract

There is a common agreement among those who deal with the theory of dynamic NMR spectrum simulations that the sum of the Hamiltonian of the whole spin system – incorporating all scalar couplings and exchange processes – and the matrix of the superoperator describing the exchange processes has to be calculated and diagonalized in the space of the single quantum coherences in order to get the spectrum. However, I have shown that it need not have to be so. Taking into account all the single quantum coherences of the whole spin system is only a necessary enlargement of the simulation space due to the fact that the exchange processes cannot be described in the space spanned by the single quantum coherences of a spin set.

During my PhD work I have shown that the effects of the exchange processes and the scalar couplings can be separated in the simulation. The spin interactions are described by quantum mechanics while the dynamic effects are characterized by statistical methods. The easiest way to handle the latter is by means of kinetic Monte Carlo simulation but there is an opportunity to evaluate a general formula without random numbers as well.

The calculation method presented here also provides a possible extension of the single spin vector model. The crucial point is the interpretation of the linear transformation between the basis functions and the eigenfunctions (or coherences) during the detection and exchange processes. These two processes can be described by the population changes of single quantum coherences and it cannot be handled using the connection between basis and eigenfunctions. However the mathematical description is complete in the Hilbert space and there is no need to calculate in the larger space spanned by the coherences.

A program, ProMoCS, was written based on the mathematical equations and it gave matching results with the spectra calculated by the former formalism in all test systems. The main advantage of the ProMoCS program is that it deals with significantly smaller matrices as compared to the former DNMR spectrum simulators. It can be parallelized to multicore processors and programmable video cards easily and effectively. The program implemented on video cards using CUDA is 30–100 times faster than the original one.

The runtime of ProMoCS scales exponentially on the size of the spin set and it is proportional to the rates of the exchange processes. On the other hand it is independent of the number of conformers which makes it very useful in the case of small spin sets with exchanges with many possible sites.

Függelék

F.1 A teszteléshez használt spinrendszerek adatai

A kisebb spinrendszereket az 1– m oszlopok az 1– n magokhoz tartozó sorai adják.

	1	2	3	4
δ_1/ppm	−0,5	1,0	−0,4	0,9
δ_2/ppm	−1,3	0,6	−1,1	0,4
δ_3/ppm	−0,6	1,2	0,4	−0,8
δ_4/ppm	−0,1	0,3	−0,3	0,5
δ_5/ppm	−0,9	0,8	−0,7	0,6
δ_6/ppm	−0,9	0,8	−0,9	0,8
δ_7/ppm	0,1	−0,4	0,6	−1,3
δ_8/ppm	−0,3	0,9	−0,3	0,9
J_{12}/Hz	10,0	4,0	10,0	4,0
J_{13}/Hz	7,0	3,0	1,0	2,0
J_{14}/Hz	0,0	0,0	0,0	0,0
J_{15}/Hz	4,0	4,0	4,0	4,0
J_{16}/Hz	4,0	4,0	4,0	4,0
J_{17}/Hz	0,0	0,0	0,0	0,0
J_{18}/Hz	5,0	5,0	5,0	5,0
J_{23}/Hz	5,0	5,0	2,0	11,0
J_{24}/Hz	7,0	4,0	7,0	4,0
J_{25}/Hz	10,0	5,0	10,0	5,0
J_{26}/Hz	10,0	5,0	10,0	5,0
J_{27}/Hz	4,0	3,0	6,0	3,5
J_{28}/Hz	0,0	8,0	0,0	8,0
J_{34}/Hz	4,0	7,0	4,0	5,0
J_{35}/Hz	5,5	10,0	5,0	6,0
J_{36}/Hz	5,5	10,0	5,0	6,0
J_{37}/Hz	3,0	4,0	3,0	4,0
J_{38}/Hz	6,0	0,0	8,0	0,0
J_{45}/Hz	0,0	0,0	0,0	0,0
J_{46}/Hz	0,0	0,0	0,0	0,0
J_{47}/Hz	7,0	2,0	2,0	2,0
J_{48}/Hz	4,3	3,1	4,3	3,1
J_{56}/Hz	0,0	0,0	0,0	0,0
J_{57}/Hz	11,0	7,0	7,0	5,0
J_{58}/Hz	2,3	2,0	2,3	2,0
J_{67}/Hz	11,0	7,0	7,0	5,0
J_{68}/Hz	2,3	2,0	2,3	2,0
J_{78}/Hz	0,0	5,0	0,0	2,0

A sebességi együtthatók a legalacsonyabb (T_1) hőmérsékleten. A többi hőmérsékleten ezen értékek 10, 100, 1000, 10000-szereseit használtuk.

h	k_{1h}	k_{2h}	k_{3h}	k_{4h}
1	–	2,0	1,5	0,0
2	1,0	–	0,0	1,5
3	1,5	0,0	–	1,5
4	0,0	1,5	1,0	–

F.2 A teszt futásidők.

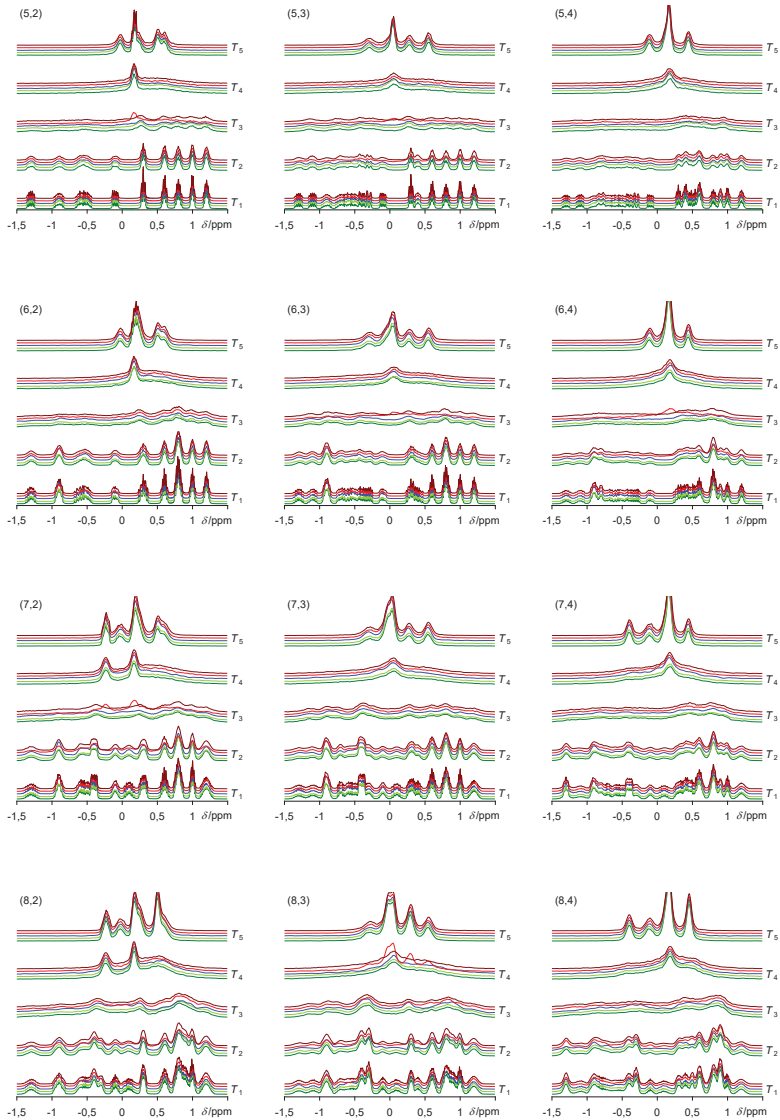
		<i>T</i>	1 mag	2 mag	3 mag	4 mag	5 mag	6 mag	7 mag	8 mag
ProMoCS-Java 1 szálon	<i>m</i> = 2	0	140 ms	578 ms	1,9 s	6,1 s	18,5 s	1,2 p	4,4 p	14,7 p
		1	125 ms	509 ms	1,6 s	5,2 s	15,7 s	1,1 p	3,9 p	14,3 p
		2	134 ms	519 ms	1,7 s	5,4 s	17,3 s	1,4 p	6,7 p	32,4 p
		3	325 ms	1,2 s	3,9 s	14,8 s	58,2 s	6,1 p	38,6 p	3,9 h
		4	2,3 s	8,5 s	27,1 s	1,8 p	8 p	53,9 p	6,1 h	1,5 d
	<i>m</i> = 3	0	141 ms	562 ms	1,9 s	6 s	18,1 s	1,2 p	4,3 p	14,4 p
		1	125 ms	500 ms	1,6 s	5,2 s	15,7 s	1,1 p	4 p	14,7 p
		2	141 ms	551 ms	1,7 s	5,8 s	18,8 s	1,6 p	7,9 p	39,1 p
		3	429 ms	1,6 s	5 s	18,5 s	1,2 p	7,8 p	51 p	5,1 h
		4	3,4 s	11,5 s	36,9 s	2,5 p	10,5 p	1,2 h	7,9 h	2,1 d
	<i>m</i> = 4	0	141 ms	569 ms	1,8 s	5,8 s	17,2 s	1,2 p	4 p	13,4 p
		1	125 ms	485 ms	1,5 s	4,9 s	14,9 s	1 p	3,9 p	15 p
		2	162 ms	611 ms	1,9 s	6,4 s	21,8 s	1,9 p	10,2 p	55,4 p
		3	621 ms	2,2 s	6,9 s	26,5 s	1,8 p	11,8 p	1,3 h	7,9 h
		4	5,2 s	17,9 s	56,9 s	3,8 p	16,4 p	1,9 h	12,5 h	3,2 d
ProMoCS-Java 2 szálon	<i>m</i> = 2	0	62 ms	297 ms	969 ms	2,6 s	9,3 s	32,8 s	2,2 p	7,1 p
		1	62 ms	266 ms	813 ms	2,6 s	7,9 s	32,6 s	2 p	7,2 p
		2	63 ms	266 ms	828 ms	2,7 s	9,1 s	41,4 s	3,3 p	16,2 p
		3	172 ms	671 ms	2,1 s	7,6 s	29,8 s	3 p	19,4 p	1,9 h
		4	1,3 s	4,5 s	14,6 s	55,8 s	4 p	26,9 p	3 h	18,7 h
	<i>m</i> = 3	0	63 ms	282 ms	953 ms	2,6 s	7,9 s	36,1 s	2,1 p	7,2 p
		1	62 ms	250 ms	797 ms	2,6 s	7,8 s	32 s	2 p	7,3 p
		2	78 ms	281 ms	859 ms	2,9 s	9,4 s	46,2 s	3,8 p	19,5 p
		3	235 ms	844 ms	2,5 s	9,4 s	37,1 s	3,9 p	25 p	2,5 h
		4	1,9 s	6,4 s	19 s	1,2 p	5,3 p	35,3 p	4 h	1 d
	<i>m</i> = 4	0	62 ms	265 ms	922 ms	2,5 s	7,5 s	35,1 s	2 p	6,7 p
		1	62 ms	234 ms	765 ms	2,5 s	7,4 s	31,2 s	2 p	7,6 p
		2	78 ms	328 ms	969 ms	3,2 s	11 s	57,1 s	5,1 p	27,7 p
		3	360 ms	1,2 s	3,7 s	13,5 s	54,9 s	5,9 p	39,1 p	4 h
		4	3 s	10 s	30,5 s	1,9 p	8,3 p	55,7 p	6,3 h	1,6 d
ProMoCS-C	<i>m</i> = 2	0	78 ms	313 ms	969 ms	3,1 s	8,9 s	36,3 s	2,1 p	6,9 p
		1	94 ms	344 ms	985 ms	3,1 s	9,1 s	37,3 s	2,2 p	7,4 p
		2	93 ms	359 ms	1,1 s	3,6 s	10,9 s	47,2 s	3,1 p	13 p
		3	225 ms	831 ms	2,5 s	8,5 s	28,6 s	2,5 p	13,5 p	1,2 h
		4	1,5 s	5,3 s	15,7 s	53,7 s	3,2 p	18,5 p	1,8 h	9,8 h
	<i>m</i> = 3	0	78 ms	328 ms	969 ms	3,1 s	8,9 s	36 s	2,1 p	7,1 p
		1	93 ms	328 ms	984 ms	3,1 s	9,2 s	37,3 s	2,3 p	7,8 p
		2	93 ms	391 ms	1,2 s	3,8 s	11,5 s	50,7 s	3,6 p	15 p
		3	290 ms	1 s	3,1 s	10,3 s	34,8 s	3,1 p	16,5 p	1,5 h
		4	1,9 s	6,9 s	20,3 s	1,1 p	4,1 p	23,6 p	2,3 h	12,5 h
	<i>m</i> = 4	0	78 ms	312 ms	953 ms	3,1 s	8,8 s	35,5 s	2 p	6,6 p
		1	94 ms	344 ms	1 s	3,2 s	9,2 s	37,6 s	2,2 p	7,7 p
		2	109 ms	422 ms	1,3 s	4,2 s	12,9 s	58,7 s	4,3 p	19,5 p
		3	422 ms	1,5 s	4,3 s	14,4 s	50 s	4,5 p	24,9 p	2,3 h
		4	2,8 s	10 s	29,9 s	1,7 p	6 p	34,5 p	3,4 h	18,6 h

		T	1 mag	2 mag	3 mag	4 mag	5 mag	6 mag	7 mag	8 mag
ProMoCS-CUDA (GeForce8600)	$m = 2$	0	16 ms	47 ms	141 ms	594 ms	2,6 s	7,0 s	24,9 s	54,7 s
		1	15 ms	31 ms	153 ms	553 ms	2,1 s	5,2 s	18 s	44,3 s
		2	16 ms	40 ms	134 ms	453 ms	1,8 s	6,9 s	34,8 s	2,0 p
		3	94 ms	203 ms	515 ms	1,4 s	5,2 s	35,2 s	4,2 p	15,8 p
		4	781 ms	1,7 s	3,9 s	9,5 s	36,2 s	5,3 p	39,8 p	2,5 h
	$m = 3$	0	16 ms	47 ms	156 ms	609 ms	2,7 s	7,0 s	24,8 s	54,7 s
		1	15 ms	40 ms	156 ms	532 ms	2,1 s	5,3 s	18,1 s	46,5 s
		2	27 ms	47 ms	185 ms	556 ms	2,0 s	8,0 s	42,2 s	2,5 p
		3	172 ms	403 ms	967 ms	2,3 s	7,2 s	45,9 s	5,4 p	20,6 p
		4	1,4 s	3,3 s	7,8 s	17,4 s	55,1 s	7,1 p	52,4 p	3,3 h
	$m = 4$	0	31 ms	47 ms	156 ms	625 ms	2,7 s	7,0 s	24,8 s	54 s
		1	16 ms	31 ms	136 ms	468 ms	1,9 s	5,0 s	17,4 s	49,2 s
		2	31 ms	81 ms	240 ms	681 ms	2,4 s	10,6 s	59,9 s	3,6 p
		3	266 ms	609 ms	1,4 s	3,3 s	10,5 s	1,2 p	8,5 p	32,1 p
		4	2,2 s	5,3 s	12,2 s	27,3 s	1,4 p	11,3 p	1,4 h	5,2 h
ProMoCS-CUDA (GTX250)	$m = 2$	0	9 ms	17 ms	34 ms	80 ms	0,45 s	1,1 s	4,4 s	11,9 s
		1	9 ms	17 ms	35 ms	80 ms	0,37 s	0,99 s	3,8 s	11,3 s
		2	8 ms	19 ms	38 ms	67 ms	0,32 s	1,2 s	6,3 s	21,8 s
		3	51 ms	114 ms	156 ms	241 ms	0,93 s	5,2 s	35 s	2,2 p
		4	385 ms	860 ms	1,1 s	1,7 s	6,3 s	43,7 s	5,2 p	20,2 p
	$m = 3$	0	9 ms	17 ms	36 ms	81 ms	0,45 s	1,1 s	4,4 s	12,1 s
		1	10 ms	18 ms	36 ms	80 ms	0,36 s	0,97 s	3,7 s	11,4 s
		2	14 ms	32 ms	53 ms	88 ms	0,37 s	1,4 s	7,3 s	25,9 s
		3	76 ms	190 ms	267 ms	404 ms	1,3 s	6,7 s	45,1 s	2,9 p
		4	534 ms	1,3 s	2,0 s	3,0 s	9,6 s	58,6 s	6,9 p	26,5 p
	$m = 4$	0	9 ms	18 ms	36 ms	83 ms	0,45 s	1,1 s	4,6 s	11,3 s
		1	9 ms	17 ms	33 ms	73 ms	0,32 s	0,91 s	3,8 s	10,7 s
		2	17 ms	45 ms	70 ms	109 ms	0,43 s	1,7 s	9,9 s	34,8 s
		3	113 ms	270 ms	384 ms	586 ms	1,8 s	10,1 s	1,2 p	4,4 p
		4	822 ms	2,0 s	3,2 s	4,8 s	14,9 s	1,5 p	10,9 p	41,8 p

F.3 A teszt spektrumok

A spektrumsorozatok felső sarkában az (n,m) párok szerepelnek. A színek jelentése: ProMoCS-Java egy (–), illetve két szálon (–), ProMoCS-C (–) és ProMoCS-CUDA (–, –).





Irodalomjegyzék

A dolgozat alapjául szolgáló publikációk

- [1] Zs. Szalay, J. Rohonczy: Monte Carlo Simulation of DNMR spectra of coupled spin systems. *J. Magn. Reson.* **191** (2008) 56–65.
- [2] Zs. Szalay, J. Rohonczy: Simulation of DNMR spectra using propagator formalism and Monte Carlo method. *J. Magn. Reson.* **197** (2009) 48–55.
- [3] Zs. Szalay, J. Rohonczy: Monte Carlo simulation of NMR lineshapes in chemically exchanging spin systems. *Progr. Nucl. Magn. Reson. Spect.* **56** (2010) 198–216.
- [4] Zs. Szalay, J. Rohonczy: Fast Calculation of DNMR Spectra on CUDA Enabled Video Card. *J. Comp. Chem.* (beküldve).

Irodalmi hivatkozások

- [5] A. Gelling, K.G. Orrell, A.G. Osborne, V. Sik, *J. Chem. Soc., Dalton Trans.* (1998) 937–945.
- [6] A.D. Bain, R.A. Bell, D.A. Fletcher, P. Hazendonk, R.A. Maharajh, S.Rigby J.F. Valliant, *J. Chem. Soc., Perkin Trans. 2* (1999) 1447–1453.
- [7] M.S. Ward, F.T. Lin, R.E. Shepherd, *Inorg. Chim. Acta* **343** (2003) 231–243.
- [8] F. Gasparrini, L. Lunazzi, A. Mazzanti, M. Pierini, K.M. Pietrusiewicz, C. Villani, *J. Am. Chem. Soc.* **122** (2000) 4776–4780.
- [9] M. Pons, O. Millet, *Progr. Nucl. Magn. Reson. Spect.* **38** (2001) 267–324.
- [10] R.M. Claramunt, C. Lopez, M.D. Santa María, D. Sanz, J. Elguero, *Progr. Nucl. Magn. Reson. Spect.* **49** (2006) 169–206.
- [11] B.M. Rode, C.F. Schwenk, T.S. Hofer, B.R. Randolph, *Coord. Chem. Rev.* **249** (2005) 2993–3006.
- [12] J. I. Kaplan, G. Fraenkel, *NMR of Chemically Exchanging Systems*, Academic Press, New York, 1980.
- [13] J.K.M. Sanders, B.K. Hunter, *Modern NMR Spectroscopy* Oxford Univ. Press, Oxford, 1993.
- [14] A. E. Derome, *Modern NMR Techniques for Chemistry Research*, Pergamon Press, Oxford, 1993.
- [15] J. Sandstrom, *Dynamic NMR Spectroscopy*, Academic Press, London, 1982.
- [16] R.R. Ernst, G. Bodenhausen, A. Wokaun, *Principles of Nuclear Magnetic Resonance in One and Two Dimensions*, Clarendon Press, Oxford 1987.
- [17] L. M. Jackman, F. A. Cotton, *Dynamic Nuclear Magnetic Resonance Spectroscopy*; Academic Press, New York, 1975.
- [18] J. W. Emsley, J. Feeney, L. H. Sutcliffe: *High Resolution NMR Spectroscopy*, Pergamon Press, New York, 1965.
- [19] Z. Luz, R. Naor, Use of group theory for calculation of dynamic N.M.R. spectra applications to liquid crystalline solutions, *Mol. Phys.* **46** (1982) 891–912.
- [20] R.S. Dumont, S. Jain, A.D. Bain, *J. Chem. Phys.* **106** (1997) 5928–5936.
- [21] R.S. Dumont, P. Hazendouk, A.D. Bain, *J. Chem. Phys.* **113** (2000) 3270–3281.

-
- [22] CUDA, http://www.nvidia.com/object/cuda_home.html
- [23] S. Chib, E. Greenberg, *Am. Stat.* **49** (1995) 327–335.
- [24] I. M. Szobol, A Monte-Carlo módszerek alapjai, Műszaki Könyvkiadó, Budapest, 1981.
- [25] A.F. Bilajev, Fundamentals of the Monte Carlo method for neutral and charged particle transport, University of Michigan, 2001.
- [26] I.T. Dimov, Monte Carlo Methods For Applied Scientists, World Scientific Publishing, Singapore, 2008.
- [27] M. Creutz, *Phys. Rev. Lett.* **50** (1983) 1411–1415.
- [28] J. Lee, *Phys. Rev. Lett.* **71** (1993) 211–215.
- [29] S. Rangelowa, L. Kulak, I. Gryczynski, P. Sakar, P. Bojarski, *Chem. Phys. Lett.* **452** (2008) 105–109.
- [30] M.S. Kelkar, J.L. Rafferty, E.J. Maginna, J.I. Siepmann, *Fluid Ph. Equil.* **260** (2007) 218–231.
- [31] T.M. Alam, *J. Non-Cryst. Solids* **274** (2000) 39–49.
- [32] F. Liang, W. H. Wong, *J. Chem. Phys.* **115** (2001) 344–351.
- [33] U.H.E. Hansmann Y. Okamoto, *Curr. Op. Struct. Biol.* **9** (1999) 177–183.
- [34] N. Metropolis, The Beginning of the Monte Carlo Method, *Los Alamos Science, Special Issue*, (1987) 125–130.
- [35] J. Sanz, C.P. Herrero, J.-L. Robert, *J. Phys. Chem. B* **107** (2003) 8337–8342.
- [36] G. Maurin, P. Senet, S. Devautour, P. Gaveau, F. Henn, V. E. Van Doren, J. C. Giuntini, *J. Phys. Chem. B* **105** (2001) 9157–9161.
- [37] M.O. Coppens, A.T. Bell, A.K. Chakraborty, *Chem. Eng. Sci.* **54** (1999) 3455–3463.
- [38] A. Lemak, C.A. Steren, C.H. Arrowsmith, M. Llinás, *J. Biomol. NMR* **41** (2008) 29–41.
- [39] D. Nathan, D.M. Crothers, *J. Mol. Biol.* **316** (2002) 7–17.
- [40] G. La Penna, A. Mitsutake, M. Masuya, Y. Okamoto, *Chem. Phys. Lett.* **380** (2003) 609–619.
- [41] E. Fieremans, Y. Deene, S. Delputte, M.S. Özdemir, Y. D’Asseler, J. Vlassenbroeck, K. Deblaere, E. Achten, I. Lemahieu, *J. Magn. Reson.* **190** (2008) 189–199.
- [42] L.F. Gladdena, M. Hargreaves, P. Alexander, *Chem. Eng. J.* **74** (1999) 57–66.
- [43] P. Porion, L.J. Michot, A.M. Faugère, A. Delville, *J. Phys. Chem. C* **111** (2007) 5441–5453.
- [44] S.K.K. Lam, H. Peemoeller, Z.Y. Chen, *J. Coll. & Interface Sci.* **248** (2002) 255–259.
- [45] D.N. Sears, R.E. Wasylishen, T. Ueda, *J. Phys. Chem. B* **110** (2006) 11120–11127.
- [46] E. Anordo, F. Grinberg, M. Vilfan, R. Kimmich, *Chem. Phys.* **297** (2004) 99–110.

- [47] M.C. Böhm, R. Ramirez, J. Schulte, *Chem. Phys.* **342** (2007) 1–15.
- [48] J.S.J. Lee, R.O. Sokolovskii, R. Berardi, C. Zannoni, E.E. Burnell, *Chem. Phys. Lett.* **454** (2008) 56–60.
- [49] C. Chiccoli, P. Pasini, G. Skacej, C. Zannoni, S. Zumer, *Phys. Rev. E* **60** (1999) 4219–4225.
- [50] A. Galindo, G. Jackson, D.J. Photinos, *Chem. Phys. Lett.* **325** (2000) 631–638.
- [51] L. Rui-Yan, Q. Jing-Yu, G. Ting-Kun, B. Xiu-Fang, *J. Non-Cryst. Solids* **354** (2008) 1736–1739.
- [52] K. Itoh, M. Sonobe, M. Sugiyama, K. Mori, T. Fukunaga, *J. Non-Cryst. Solids* **354** (2008) 150–154.
- [53] M. Fábán, E. Sváb, Th. Proffen, E. Veress, *J. Non-Cryst. Solids* **354** (2008) 3299–3307.
- [54] K.D. Machadoa, D.F. Sanchez, J.C. de Limab, T.A. Grandi, *Sol. St. Comm.* **148** (2008) 46–49.
- [55] S. Mudry, I. Shtablavyi, V. Sklyarchuk, Y. Plevachuk, *J. Nucl. Mat.* **376** (2008) 371–374.
- [56] F. Utsuno, H. Inoue, Y. Shimane, T. Shibuya, K. Yano, K. Inoue, I. Hirose, M. Sato, T. Honma, *Thin Solid Films* **516** (2008) 5818–5821.
- [57] W.K. Luo, E. Ma, *J. Non-Cryst. Solids* **354** (2008) 945–955.
- [58] J.C. McLaughlin, S.L. Tagg, J.W. Zwanziger, D.R. Haefner, S.D. Shastri, *J. Non-Cryst. Solids* **274** (2000) 1–8.
- [59] L. Pusztai, I. Harsányi, H. Dominguez, O. Pizio, *Chem. Phys. Lett.* **457** (2008) 96–102.
- [60] M.H. Sørby, A. Møllergaard, B.C. Hauback, H. Fjellvåg, R.G. Delaplane, *J. Alloys & Comp.* **457** (2008) 225–232.
- [61] S. Takeda, H. Fujii, Y. Kawakita, S. Tahara, S. Nakashima, S. Kohara, M. Itou, *J. Alloys & Comp.* **452** (2008) 149–153.
- [62] M. Wilding, Y. Badyal, A. Navrotsky, *J. Non-Cryst. Solids* **353** (2007) 4792–4800.
- [63] J.C. McLaughlin, J.W. Zwanziger, *J. Mol. Graph. & Mod.* **17** (1999) 275–284.
- [64] F.L. Tobiason, G. Vergotenb, J. Mazuricr, *J. Mol. Struct. (Theochem)* **395–396** (1997) 173–185.
- [65] P.H. He, Y. Wu, R.F. Wu, *Curr. App. Phys.* **7S1** (2007) e63–e67.
- [66] J.D. Beck, B.H. Meier, H. Schäfer, *J. Magn. Reson.* **162** (2003) 141–157.
- [67] T.K. Hitchens, J.A. Lukin, Y. Zhan, S.A. McCallum, G.S. Rule, *J. Biomol. NMR*, **25** (2003) 1–9.
- [68] D.V. Rubtsov, J.L. Griffin, *J. Magn. Reson.* **188** (2007) 367–379.
- [69] A. Ebel, W. Dreher, D. Leibfritz, *J. Magn. Reson.* **182** (2006) 330–338.
- [70] R. Romano, A. Motta, S. Camassa, C. Pagano, M.T. Santini, P.L. Indovina, *J. Magn. Reson.* **155** (2002) 226–235.

- [71] R. Romano, D. Paris, F. Acernese, F. Barone, A. Motta, *J. Magn. Reson.* **192** (2008) 294–301.
- [72] I. G. Darvey, P. J. Staff, *J. Chem. Phys.* **44** (1966) 990–997.
- [73] D.T. Gillespie, *J. Comp. Phys.* **22** (1976) 403–434.
- [74] D.T. Gillespie, *J. Chem. Phys.* **113** (2000) 297–306.
- [75] M. Rathinam, R. L. Petzold, Y. Cao, D.T. Gillespie, *J. Chem. Phys.* **119** (2003) 12784–12794.
- [76] A. Slepoy, A. P. Thompson, S. J. Plimpton, *J. Chem. Phys.* **128** (2008) 205101.
- [77] D.T. Gillespie, *J. Chem. Phys. A* **106** (2002) 5063–5071.
- [78] B. H. Robinson, A. W. Reese, E. Gibbons, C. Mailer, *J. Phys. Chem. B* **103** (1999) 5881–5894.
- [79] M. Carl, G.W. Miller, J.P. Mugler III, S. Rohrbaugh, W.A. Tobias, G.D. Cates Jr., *J. Magn. Reson.* **189** (2007) 228–240.
- [80] C. Caia, Z. Chena, S. Caia, J. Zhong, *J. Magn. Reson.* **172** (2005) 242–253.
- [81] S. Vasenkov, J. Kärger, D. Freude, R.A. Rakoczy, J. Weitkamp, *J. Mol. Cat. A* **158** (2000) 373–376.
- [82] D.S. Grebenkov, G. Guillot, B. Sapoval, *J. Magn. Reson.* **184** (2007) 143–156.
- [83] D. Idiyatullin, S. Michaeli, M. Garwood, *J. Magn. Reson.* **171** (2004) 330–337.
- [84] E.L. Kovrigin, J.G. Kempf, M.J. Grey, J.P. Loria, *J. Magn. Reson.* **180** (2006) 93–104.
- [85] M. Harris, Mapping computational concepts to GPUs. In ACM SIGGRAPH 2005 Courses. Editor: J. Fujii. ACM Press, New York, NY, 50.
- [86] AMD FireStream, <http://developer.amd.com/gpu/ATIStreamSDK>
- [87] Apple Snow Leopard, <http://www.apple.com/macosx/developers/>
- [88] Khronos Group, <http://www.khronos.org/OpenGL/>
- [89] M. C. Schatz, C. Trapnell, C. Delcher, A. L. Varshney, *BMC Bioinformatics* **8** (2007) 474.
- [90] J. E. Stone, J. C. Phillips, P. L. Freddolino, D. J. Hardy, L. G. Trabuco, K. Schulten, *J. Comp. Chem.* **28** (2007) 2618.
- [91] M. Moorkamp, M. Jegen, A. Roberts, R. Hobbs, *Comp. & Geosci.* **36** (2010) 680–686.
- [92] S. Scarle, *Comp. Biol. & Chem.* **33** 253–260.
- [93] E. Alerstam, T. Svensson, S. Andersson-Engels, *J. Biomed. Opt.* **13** (2008) 060504.
- [94] T. Narumi, K. Yasuoka, M. Taiji, S. Höfinger, *J. Comp. Chem.* **30** (2009) 2351–2357.
- [95] M. S. Friedrichs, P. Eastman, V. Vaidyanathan, M. Houston, S. Legrand, A. L. Beberg, D.L. Ensign, C.M. Bruns, V.S. Pande, *J. Comp. Chem.* **30** (2009) 864–872.
- [96] T. Preis, P. Virnau, W. Paula, J. J. Schneidera, *J. Comp. Phys.* **228** (2009) 4468–4477.

- [97] N. Sanna, I. Baccarelli, G. Morelli, *Comp. Phys. Comm.* **180** (2009) 2544–2549.
- [98] P. M. Kasson, D. L. Ensign, V. S. Pande, *J. Am. Chem. Soc.* **131** (2009) 11338–11340.
- [99] D. S. Tarasov, E.D. Izotova, D. A. Alisheva, N. I. Akberova, *Mat. Models & Comp. Sim.* **2** (2010) 46–54.
- [100] S. Ufimtsev, T.J. Martínez *J. Chem. Theory Comput.* **4** (2008) 222–231.
- [101] K. Yasuda, *J. Comp. Chem.* **29** (2008) 334–342.
- [102] L. Vogt, R. Olivares-Amaya, S. Kermes, Y. Shao, C. Amador-Bedolla, A. Aspuru-Guzik, *J. Phys. Chem A* **112** (2008) 2049–2057.
- [103] A. G. Anderson, W. A. Goddard, P. Schröde, *Comp. Phys. Comm.* **177** 298–306.
- [104] M. Aoki, H. Tomono, T. Iitaka, K. Tsumuraya, *J. Phys.: Conf. Ser.* **215** (2010) 012120.
- [105] L. Petzold, H. Li, *Int. J. High Perf. Comp. App.* **24** (2010) 110–116.
- [106] I. J. Cutress, R. G. Compton, *J. Electroanal. Chem.* **643** 102–109.
- [107] F. Molnár Jr., T. Szakály, R. Mészáros, I. Lagzi, *Comp. Phys. Comm.* **181** 105–112.
- [108] S. A. Manavski, G. Valle, *BMC Bioinformatics* **9** (2008) S10.
- [109] C. Trapnell, M. C. Schatz, *Paral. Comp.* **35** 429–440.
- [110] R. Hussong, B. Gregorius, A. Tholey, A. Hildbrant, *Bioinformatics* **25** (2009) 1937–1943.
- [111] CUDA 2.3 Programming Manual, http://developer.download.nvidia.com/compute/cuda/2.3/toolkit/docs/NVIDIA_CUDA_Programming_Guide_2.3.pdf
 CUDA 2.0 Programming Manual http://developer.download.nvidia.com/compute/cuda/2.0/docs/NVIDIA_CUDA_Programming_Guide_2.0.pdf
- [112] Sohár P. Mágneses magrezonancia spektroszkópia, Akadémiai Kiadó, Budapest, 1976.
- [113] G. Binsch, *J. Am. Chem. Soc.* **91** (1969) 1304–1309.
- [114] R. Kubo, *J. Phys. Soc. Jpn.* **9** (1954) 935.
- [115] R.A. Sack, *Mol. Phys.* **1** (1958) 163.
- [116] A.G. Redfield, *Adv. Magn. Reson.* **1** (1965) 1.
- [117] A. D. Bain, *Conc. Magn. Reson. A* **10** (1998) 85–98.
- [118] A.D. Bain, G. J. Duns, *Can. J. Chem.* **74** (1996) 819–824.
- [119] S. Szymanski, G. Binsch, *J. Magn. Reson.*, **81** (1989) 104–120.
- [120] D. S. Stephenson, G. Binsch, *J. Magn. Reson.* **32** (1978) 145–152.
- [121] D. A. Kleier, G. Binsch, *J. Magn. Reson.* **3** (1970) 146–160.
- [122] D. S. Stephenson, G. Binsch, *QCPE* **11** 365 (1978)
- [123] A. D. Bain, *Progr. Nucl. Magn. Res.* **43** (2003) 63–103.
- [124] A.D. Bain, D.M.Rex, R.N. Smith, *Magn. Reson. Chem.* **39** (2001) 122–126.
- [125] A.D. Bain, G.J. Duns, *J. Magn. Reson. A* **112** (1995) 258–260.

- [126] A. D. Bain: MEXICO3 (2002)
Letölthető: <http://www.chemistry.mcmaster.ca/bain/exchange.html>
- [127] H.J. Reich, W.S. Goldenberg, B.Ö. Gudmundsson, A.W. Sanders, K.J. Kulicke, S. Simon, I.A. Guzei, *J. Am. Chem. Soc.* **123** (2001) 8067–8079.
- [128] H.J. Reich, *J. Chem. Educ. Software* 3D2 (1996)
- [129] Reich, H. J. *J. Chem. Educ.* **72** (1995) 1086.
- [130] H. Reich: WINDMR 7.1 (2006).
Letölthető: <http://www.chem.wisc.edu/areas/reich/plt/windnmr.htm>
- [131] J. Rohonczy, DNMR Lineshape Analysis software manual, Version 1.1, Rev. 071103, Bruker BioSpin GmbH, TopSpin 2.1 package
- [132] J. Rohonczy, *Kem. Kozl.* **74** (1992) 161–200.
- [133] MNova-NMR (korábban MestreC) (2010).
Honlap: <http://mestrelab.com/Products/Mnova-NMR/Details.html>
- [134] iNMR (2006). Honlap: www.inmr.net
- [135] K. Marat: SpinWorks (2004). <http://www.umanitoba.ca/chemistry/nmr/spinworks>
- [136] NUTS (2010): <http://www.acornnmr.com/nuts.htm>
- [137] V.F. Galat, A.N. Vdovichenko, A.Y. Chervinskii, L.M. Kapkan, *Theor. & Exp. Chem.* **31** (1995) 105
- [138] M. Cuperlovic, G.H. Meresi, W.E. Palke, J.T. Gerig, *J. Magn. Reson.* **142** (2000) 11–23.
- [139] M. Mehring, High resolution NMR in Solids, Springer-Verlag, Berlin, 1983.
- [140] N. J. Higham Functions of matrices: theory and computation, SIAM, 2008.
- [141] R. C. Ward, *SIAM J. Numer. Anal.* **14** (1977) 600–610.
- [142] I. Gay: SPIN. Letölthető: <http://www.sfu.ca/~gay>.
- [143] J. Rohonczy, D. Knausz, B. Csákvári, P. Sohár, I. Pelczer, L. Párkányi, *J. Orgmet. Chem.* **340** (1988) 293–302.
- [144] S.S. Rigby, H.K. Gupta, N.H. Werstiuk, A.D. Bain, M.J. McGlinchey, *Inorg. Chim. Acta* **251** (1996) 355–364.
- [145] A. Deák A, T. Megyes, G. Tárkányi, P. Király, L. Biczók, G. Pálincás, P.J. Stang, *J. Am. Chem. Soc.* **128** (2006) 12668.
- [146] G. Tárkányi, P. Király, G. Pálincás, A. Deák, *Magn. Res. Chem.* **45** (2007) 917.